BASIC

SOFTWARE

LIBRARY

VOLUME III

ADVANCED

BUSINESS

by R. W. BROWN

BASIC

SOFTWARE

LIBRARY

VOLUME III

ADVANCED

BUSINESS

THIS BASIC SOFTWARE LIBRARY IS MADE AVAIL
ABLE, FREE OF RESTRICTIONS AND ROYALTIES
TO SCHOOLS, COLLEGES, UNIVERSITIES, INDI-
VIDUALS, HOBBYIST & BUSINESS CONCERNS FOR
USE ON THEIR OWN COMPUTERS AND OR COMPUT-
ING SYSTEMS. REPRODUCTION IN ANY PART OR
FORM OF THIS ENTIRE LIBRARY IS STRICTLY
FORBIDDEN. USE OF ANY PART OR FORM OF THIS
ENTIRE LIBRARY FOR COMMERCIAL USE OF ANY
KIND IS STRICTLY FORBIDDEN WITHOUT THE EX
PRESSED WRITTEN PERMISSION OF SCIENTIFIC
RESEARCH.

INTRODUCTION

The programs presented here are set out for the individual who has a specific need in mind. Because a detailed discussion of these programs would require a text several times the present size of this Library it has been omitted. Individuals who have a specific requirement will have to be at least knowledgeable in the area the program is written about; ie: Statistical programs require the user to be familiar with the terms mean, median, etc. This is because the programs are written in the vernacular of their subject matter. With this knowledge alone, no programming experience on the part of the user is required in order to use any of these programs in most systems. Once it is determined that a particular program may be useful the user merely types in a copy of the BASIC source code exactly as it appears in the program listing. Then follow the instructions for running the program as presented in the Instruction portion of the write up, immediately preceding the program. Also included in the write ups are statements that appear in the source code which may possibly need to be changed to run in the user's computer system; ie: RND statements may have to be changed to FRAND in order to compile in certain systems.

PUBLISHERS NOTE: Appendix B included at the end of Volume V was not
mentioned in the preface by the author. We feel this appendix is
the most important single item included in this library. We see
this appendix as a fore runner that might lead the way toward
standardizing a computer language among the manufacturers. This is
in addition to the obvious benefits to all users of this Basic
Software Library.

TABLE OF CONTENTS

TABLE of CONTENTS

VOLUME ONE (CONT.)

PICTURES

TABLE of CONTENTS

VOLUME TWO

Part 3 - Math & Engineering Programs

TABLE of CONTENTS

VOLUME TWO (CONT.)

Part 4 - Plotting & Statistics Programs

APPENDIX A - BASIC STATEMENT DEFINITIONS

TABLE of CONTENTS

VOLUME THREE   PAGE

Part 5 - Advanced Business Programs

TABLE OF CONTENTS

VOLUME FOUR

General Purpose Programs

TABLE OF CONTENTS

VOLUME FIVE

Experimenter's Programs

TABLE of CONTENTS

VOLUME FIVE (CONT.)

Experimenter's Programs

APPENDIX B - STATEMENT CONVERSION ALGORITHMS

TABLE of CONTENTS

VOLUME SIX

A Complete Business System

x

PREFACE

The information contained in these pages represents the culmination of a
very large project. That of compiling a versatile and complete Software
Library that will be of use to a large number of diverse individuals. The
programs presented here when combined in a system will represent a very
powerful library bank. Such a work as this has been attempted in the past
in such areas as cook books, electronic source books, mathematical tables
and even computer games. But to date such a collection as this has yet to
be offered to the average individual to use as he chooses. The word "at-
temped" was used as no work is ever considered complete by everyone regar
dless of its thoroughness.

The programs presented here were chosen for their uniqueness and general
usefulness. There should be at least one program included that will be of
use to every type of individual whether they have access to a computer or
not. Computers are a wonderful and very useful tool. Through this Library
I hope to interest more people into becoming involved with computers. The
Library is written so that little or no computer programming experience
is required to invoke any of the programs. The programs that are present-
ed here are all written in the computer language called BASIC. Each pro-
gram has been successfully run on a G.E. 635 computer. The entire source
code is presented as well as a short narrative page which defines the pro
gram, tells who might be interested in using it, a brief set of instruc-
tions or how to get them and then any limitations in the program are no-
ted. In the limitations section the storage length in K Bytes is given so
the prospective user will know how much memory to allow for the program.
Where possible the amount of memory space required for full execution is
given for the programs, this space is independent of the space already oc
cupied by your BASIC compiler.

The programs are broken down into five sections or parts. Each part deals
with a specific type of program. Part 1 contains business type programs.
These programs will be of interest to individuals who have businesses,
play the stock market, balance their own checkbooks, do installment buy-
ing, figure taxes, etc. There are a total of 20 programs in this section.
Part 2 is the lighter side of the Library as it contains 16 games and 12
picture programs. No computer library is complete without some fun. Among
the games presented in this section is one called Checkers. The game is
rather long but it is virtually machine independent as it doesn't use over
lay techniques or use files. Most of the other games included here are as
exciting as this version of Checkers. Each was chosen so as not to mimic
others that the reader may have seen. The pictures are as unusual in their
own way as are the games. Most of the pictures are spread over several pa
ges, this was done not only so the reader will need to run the program to
see the details of a particular picture but also in the hopes of getting
as many of these programs into use as possible. As the picture programs
are very simple it is an easy place for the novice to start learning about
programming.

Part 3 is comprised of Math and Engineering programs. Some of these programs
will be of use to high school students, professional people, sailors, engi-
neers, astronomers, airplane pilots, etc. Most of these programs are very

technical but they can perform every day calculations quickly and easily and they are extremely simple to use. There are 23 general usage programs presented in this section.

Part 4 is made up of Plotting and Statistical Analysis programs. These programs can be readily utilized by a number of people in widely different disciplines from fishermen to statisticians. The data gathered may be from a poll, a census, a test sample or even the number of fish caught on various days. The stat programs will be of invaluable aid to anyone who gathers data of any kind. The plotting routines will be of use to most of the people who use the stat programs or programs in Parts 1 and 3. The plotting is done on any standard teletype or terminal and does not require a special plotter or plotting terminal. There are a total of five direct plotting programs and 18 stat programs in this section.

All of the programs presented here may be run by simply typing the source code as listed, exactly as it is, into your computer. Now before the program will run it will have to be converted into machine code. This is done automatically and requires no forethought except to make certain the operating system you are working in is BASIC. In the larger computer systems you are asked what system you want — to this type BASIC; the smaller systems only have BASIC, in these you are O.K.

Immediately following Part 4 is Appendix A. Here, all of the Basic Statements used throughout these pages are defined. Each statement is explained sufficiently well to enable one unfamiliar with this subset to modify any necessary statements so that the program or programs will compile and execute with the Basic compiler or interpreter available with their particular computer. Most of the Basic compilers available today, that require more then 10K Bytes of storage, will execute all of the programs presented in these volumes with the possible exception of a few of the games and the program "Variable". Multiple line statements are not used in most of the programs and only a few programs use string manipulations extensively. A few of the programs may require more on line storage then is available on some of the small micro computer systems; these longer programs will not be executable due to the limited amount of memory. However most of the programs will execute in 10K Bytes of memory or less, thereby making most of the programs in this Library executable in virtually any Basic speaking computer without any required modifications.

Volume III is comprised of ADVANCED BUSINESS programs, part 5. This volume as well as subsequent volumes are intended to make this Library complete and useful to all individuals.

Each of these programs are written in a subset of the Dartmouth language. The specific subset is that which was used by General Electric on their 635 systems. These programs have operated without problem on a variety of small and large machines even several of the new micro computers. The programs that use string manipulations may require slight modifications before fully executing on some systems. These programs are mainly found in Part 2 — Games.

All of the programs in this Library were written or edited by the author.
All of the programs edited by him were given for inclusion, "swapped" for
traded, or made public. A few of the original authors of the "swaps" are
not known, for this I apologize. The others, unless specifically mentioned
in the text, are presented here. In addition I would like to thank the fol
lowing for their cooperation in making this work possible.

ACKNOWLEDGMENTS

MY WIFE MARY AND MY FAMILY

DONALD ALVAREZ	GE TIMESHARING
DAVE BEETLE	BILL JONES
MORTON BERGER	GEORGE LONG
COPY CAT INC	TOM ROSE

ARTWORK COURTESY OF   MELISSA
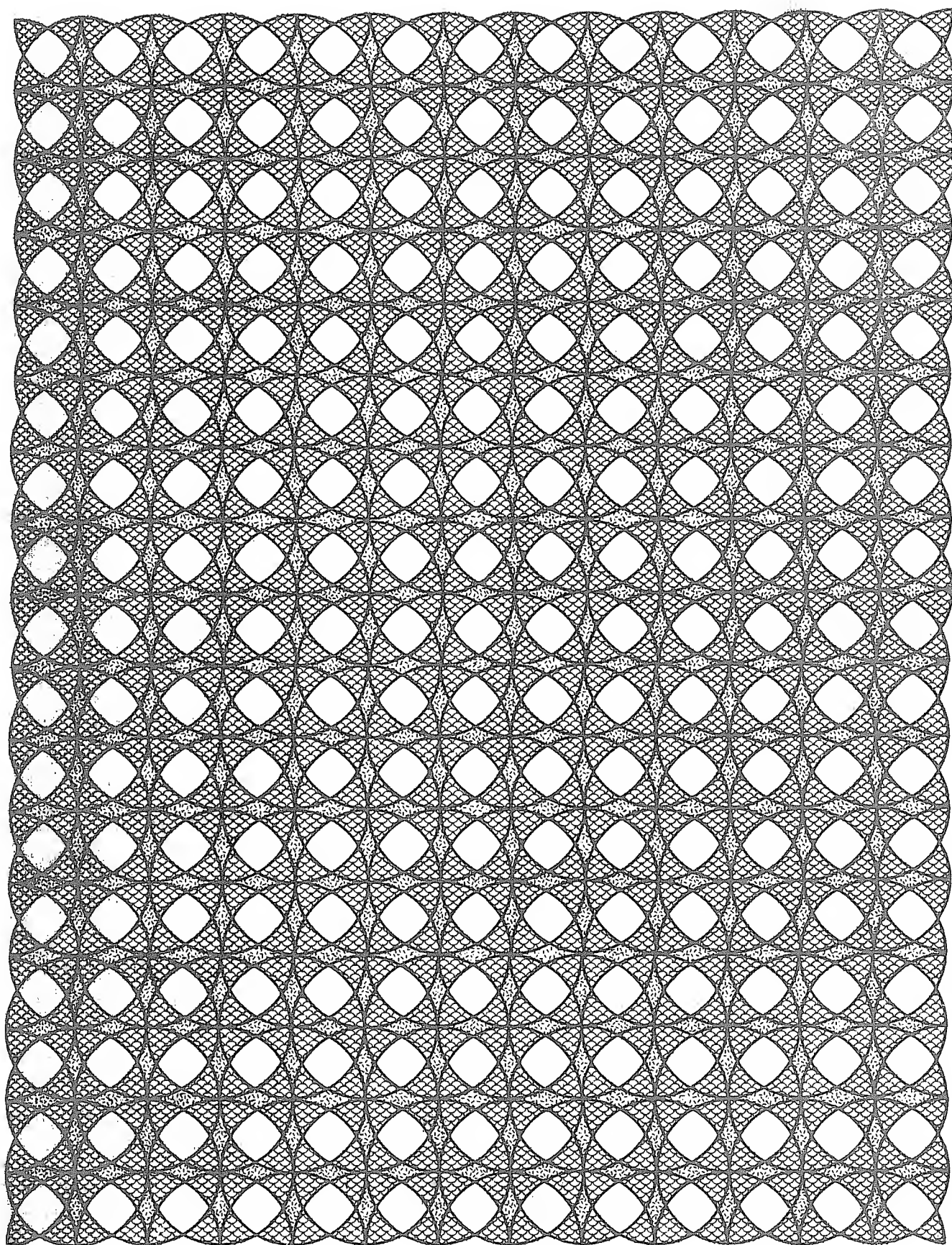
PART 5

ADVANCED BUSINESS

PROGRAMS

BILLING:

DESCRIPTION

This is a Billing and Accounts Receivable program. It does not use any external data files. All of the data is self contained within the program. It generates the following five kinds of reports: Mailing labels, Customer bills, A/R, Sales and a last purchase report.

USERS

This program would be used by individuals or companies engaged in selling merchandise or services.

INSTRUCTIONS

All of the customer and billing data must be updated in data statements before the program is run. Initially the data must be input to the program, then, after the first execution, the program with the data included is saved on paper or magnetic tape. After this the data only needs to be updated as it changes. List the program for detailed information about data entry and updating.

LIMITATIONS

As this program does not use Use Files it should execute in any Basic speaking computer that has sufficient on line storage. The program is set up with sample data that must be removed before entering your data. It is set for a maximum of 100 customers. This number is set in the DIM statements on lines 1100 and 1120. The source code requires 12K Bytes of memory for program storage. The amount of memory required for execution is a function of the number of customers you have. This is set in the DIM statements. With it set for 100 customers the program will require about 31K Bytes for execution. Included after the source code listing is an executed run using the data presently contained in the program.

```
100 REM THIS IS   THE   BILLING AND A/R PROGRAM
120 REM FOR FULL MODIFICATIONS LIST THE PROGRAM
140 REM THIS PROGRAM IS WRITTEN IN BASIC
160 REM IT DOES NOT USE EXTERNAL USE FILES
180 REM ALL UPDATING IS TO BE DONE BEFORE EXECUTION
200 REM CREATED BY SCIENTIFIC RESEARCH INST.
220 REM ALL   CUSTOMER  DATA IS CONTAINED IN THE DATA STATEMENTS
240 REM EACH TIME THE DATA IS UPDATED TO INCORPORATE CHANGES
260 REM IN  CUSTOMER   STATUS A NEW COPY OF THE PROGRAM SHOULD
280 REM BE MADE. EITHER ON TAPE OR PAPER THEN
300 REM EACH TIME THE PROGRAM IS RUN THE NEW COPY
320 REM WILL BE USED TO INPUT THE PROGRAM TO
340 REM THE COMPUTER, THE PROGRAM IS SELF PROMPTING
360 REM AND WILL ASK ALL NECESSARY QUESTIONS.
380 REM THE SUBSCRIPT (100) USED IN THE DIM
400 REM STATEMENTS IS THE MAXIMUN NUMBER OF
420 REM CUSTOMERS THIS PROGRAM IS SET UP TO HANDLE
440 REM TO INCREASE THIS NUMBER TO ANY OTHER NUMBER
460 REM SIMPLY CHANGE THIS NUMBER TO THE DESIRED CUSTOMER
480 REM LEVEL IN EACH OF THE DIM STATEMENTS INITALLY
500 REM BEFORE THE DATA IS INPUT TO THE DATA STATMENTS.
520 REM THE DATA STATEMENTS ARE USED TO UPDATE THE
540 REM A/R EVERY PERIOD. WHATEVER THIS PERIOD HAPPENS TO
560 REM BE. THE DATA STATEMENTS ARE ARRANGED IN FOUR (4)
580 REM LINE ORDERS FOR EASE OF DATA ENTRY AND UPDATING.
600 REM THE FIRST LINE CONTAINS RELATIVELY UNCHANGING DATA
620 REM LINE TWO CONTAINS PAYMENT INFORMATION
640 REM LINE THREE CONTAINS THE NUMBER OF SALES FOR A CUSTOMER
660 REM LINE FOUR CONTAINS SALES DATA - ONE SALE ON EACH LINE
680 REM        THE NUMBER OF THESE SALES LINES MUST BE CONTAINED
700 REM        IN DATA LINE NUMBER THREE OR THE PROGRAM WILL NOT RUN.
720 REM THE FOLLOWING VARIABLES ARE USED IN THIS PROGRAM
740 REM N1=NUMBER OF CUSTOMERS
760 REM D$=REPORT DATE
780 REM N(I)=ITEM #
800 REM E$(I)=ITEM DESCRIPTION (11 CHARACTERS OR LESS PLEASE)
820 REM A(I)=CUSTOMER ACCOUNT #
840 REM N$(I)=CUSTOMER NAME (20 CHARACTERS OR LESS)
860 REM R$(I)=CUSTOMER STREET ADDRESS
880 REM S$(I)=CUSTOMER CITY & STATE ADDRESS
900 REM S(I)=QUANTITY SHIPPED
920 REM A$(I)=DATE SHIPPED
940 REM       EX:   12/17/75
960 REM B(I)=# OF ITEMS PURCHASED BY CUSTOMER
980 REM P(I)=SELLING PRICE (LESS SALES TAX)
1000 REM U(I)=CUMULATIVE AMOUNT PAID ON PRESENT SALES
1020 REM U$(I)=DATE OF LAST PAYMENT
1040 REM       EX:   12/21/75
```

```
1060 REM
1080 REM    *※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*
1100 DIM N(100),E$(100),A(100),N$(100),R$(100),S$(100),S(100)
1120 DIM A$(100),B(100),P(100),U(100),U$(100)
1140 REM CHANGE THE ABOVE SUBSCRIPTS TO EQUAL YOUR NUMBER OF CUSTOMERS
1160 REM    *※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*※*
1180 N1=5
1200 REM CHANGE THE ABOVE N1 # TO EQUAL THE NUMBER OF CUSTOMERS
1220 REM THAT ARE ENTERED IN YOUR DATA STATEMENTS.
1240 REM ------------------------------------------------------------
1260 S=.04
1280 REM S IS THE SALES TAX PERCENTAGE
1300 PRINT
1320 PRINT
1340 PRINT
1360 PRINT
1380 PRINT
1400 PRINT "       THIS IS THE BILLING AND ACCOUNTS RECEIVEABLE PROGRAM "
1420 PRINT
1440 PRINT
1460 PRINT
1480 PRINT "TYPE IN TODAYS DATE, AS: 1/14/75 ";
1500 INPUT D$
1520 FOR I=1 TO N1
1540 READ A(I),N$(I),R$(I),S$(I),U(I),U$(I),B(I)
1560 IF B(I)=0 THEN 1640
1580 FOR I1=1 TO B(I)
1600 READ E$(I),N(I),A$(I),S(I),P(I)
1620 NEXT I1
1640 NEXT I
1660 PRINT
1680 PRINT
1700 PRINT
1720 PRINT
1740 PRINT
1760 PRINT
1780 PRINT
1800 PRINT "THE FOLLOWING IS A LIST OF REPORTS THAT THIS PROGRAM"
1820 PRINT "WILL GENERATE. TO CHOOSE ONE TYPE IT'S NUMBER WHEN ASKED."
1840 PRINT
1860 PRINT "     1 - PRINT MAILING LABELS "
1880 PRINT "     2 - PRINT UP BILLS "
1900 PRINT "     3 - CUSTOMER A/R REPORT "
1920 PRINT "     4 - SALES REPORT "
1940 PRINT "     5 - LAST PURCHASE REPORT "
1960 PRINT "     6 - STOP PROGRAM "
1980 PRINT
2000 PRINT "WHICH REPORT WOULD YOU LIKE TO RUN: ";
2020 INPUT R
2040 PRINT
2060 IF R=6 THEN 8640
```

```
2080 IF R=1 THEN 2200
2100 IF R=2 THEN 2720
2120 IF R=3 THEN 4480
2140 IF R=4 THEN 5200
2160 IF R=5 THEN 6340
2180 GOTO 1860
2200 REM THIS SUB GENERATES THE MAILING LIST
2220 PRINT
2240 PRINT " THIS IS THE MAILING LIST ROUTINE "
2260 PRINT
2280 PRINT "WHEN YOU HAVE THE LABELS IN PLACE TYPE A 7: ";
2300 INPUT Z2
2320 Z1=5
2340 FOR I=1 TO Z1
2360 PRINT
2380 NEXT I
2400 LET Z=3
2420 FOR I=1 TO N1
2440 PRINT TAB(Z);A(I)
2460 PRINT TAB(Z);N$(I)
2480 PRINT TAB(Z);R$(I)
2500 PRINT TAB(Z);S$(I)
2520 PRINT
2540 PRINT
2560 PRINT
2580 NEXT I
2600 PRINT
2620 PRINT
2640 PRINT
2660 PRINT "END OF THE MAILING LIST ***** "
2680 PRINT
2700 GOTO 1660
2720 REM THIS SUB GENERATES THE BILLING LIST #2
2740 B1=0
2760 B2=0
2780 B3=0
2800 REM THE FORMAT OF THE BILL PRINTING IS AS FOLLOWS:
2820 REM PAGE ONE PRINTS THE MAILING ADDRESS
2840 REM                                    DATE
2860 REM       ACCOUNT#
2880 REM       CUSTOMER NAME
2900 REM       STREET ADDRESS
2920 REM       CITY   STATE
2940 REM
2960 REM
2980 REM
3000 REM    ITEM#    DESCRIPTION   QUANTITY   DATE SHIPPED    UNIT PRICE
3020 REM       ...........     ...
3040 REM       ....         ......   ..
3060 REM
3080 REM
```

```
3100 REM TOTAL AMOUNT PAID   = $ XXX.XX
3120 REM DATE OF THE LAST PAYMENT   12/21/75
3140 REM TOTAL PIECES SHIPPED      XXXX
3160 REM TOTAL SALES TAX ON SALE = $ XX.XX
3180 REM TOTAL AMOUNT DUE    =   ******** $ XXX.XX ********
3200 REM
3220 PRINT
3240 PRINT
3260 PRINT "THIS GENERATES THE PRINTING OF THE BILLS"
3280 PRINT "WHEN THEY ARE IN PLACE TYPE A 7 :";
3300 INPUT Z
3320 LET Z=7
3340 FOR I=1 TO Z
3360 PRINT
3380 NEXT I
3400 LET Z=18
3420 RESTORE
3440 FOR I=1 TO N1
3460 LET Z2=65
3480 LET Z=5
3500 READ A(I),N$(I),R$(I),S$(I),U(I),U$(I),B(I)
3520 IF B(I)=0 THEN 4300
3540 PRINT TAB(Z);A(I)
3560 PRINT TAB(Z);N$(I)
3580 PRINT TAB(Z);R$(I)
3600 PRINT TAB(Z);S$(I)
3620 LET Z1=4
3640 FOR I3=1 TO Z1
3660 PRINT
3680 NEXT I3
3700 REM THIS IS TO SKIP DOWN TO THE STATEMENT SECTION OF THE BILL.
3720 Z7=65
3740 PRINT TAB(Z7);D$
3760 PRINT
3780 FOR I1=1 TO B(I)
3800 READ E$(I),N(I),A$(I),S(I),P(I)
3820 Z2=3
3840 PRINT TAB(Z2);
3860 PRINT N(I),E$(I),S(I),A$(I),P(I)
3880 B1=B1+S(I)
3900 B2=B2+S(I)*P(I)
3920 NEXT I1
3940 PRINT
3960 PRINT "TOTAL AMOUNT PAID       = $";U(I)
3980 PRINT "DATE OF LAST PAYMENT   =   ";U$(I)
4000 PRINT "TOTAL PIECES SHIPPED   =   ";B1
4020 B3=B2*S
4040 B3=INT(.5+100*B3)/100
4060 PRINT "TOTAL SALES TAX AMT.   = $";B3
4080 B2=B2+B3-U(I)
4100 B2=INT(.5+100*B2)/100
```

```
4120 PRINT "TOTAL AMOUNT DUE         =   **********   $";B2;"   **********  "
4140 LET Z5=10-B(I)
4160 FOR I6=1 TO Z5
4180 PRINT
4200 NEXT I6
4220 REM THIS IS TO SKIP TO THE NEXT BILL HEAD.
4240 B1=0
4260 B2=0
4280 B3=0
4300 NEXT I
4320 Z=12
4340 FOR I7=1 TO Z
4360 PRINT
4380 NEXT I7
4400 PRINT "END OF THE BILL PRINTING SEQUENCE."
4420 PRINT
4440 PRINT
4460 GOTO 1660
4480 REM THIS SUB GENERATES THE CUST. A/R REPORT #3
4500 LET B1=0
4520 LET B2=0
4540 LET B3=0
4560 PRINT
4580 PRINT
4600 PRINT "    C U S T O M E R S    A / R    R E P O R T   ";D$
4620 PRINT
4640 PRINT
4660 PRINT " ACC# ","  A/R $ "," PAYMENTS","  LS DATE"," CUST NAME"
4680 PRINT
4700 RESTORE
4720 FOR I=1 TO N1
4740 READ A(I),N$(I),R$(I),S$(I),U(I),U$(I),B(I)
4760 IF B(I)=0 THEN 5020
4780 FOR I1=1 TO B(I)
4800 READ E$(I),N(I),A$(I),S(I),P(I)
4820 T2=S(I)*P(I)
4840 T2=T2*(1+S)
4860 T2=INT(.5+100*T2)/100
4880 B3=B3+T2
4900 NEXT I1
4920 B3=INT(.5+100*B3)/100
4940 B1=B1+B3-U(I)
4960 B2=B2+U(I)
4980 PRINT A(I),B3-U(I),U(I),U$(I);"   ";N$(I)
5000 LET B3=0
5020 NEXT I
5040 PRINT
5060 PRINT
5080 PRINT
5100 PRINT "TOTAL PAYMENTS      = $";B2
5120 PRINT "TOTAL ACCOUNTS/REC. = $";B1
```

```
5140 PRINT
5160 PRINT
5180 GOTO 1660
5200 REM THIS SUB GENERATES THE SALES REPORT BY DATE #4
5220 LET B1=0
5240 LET B2=0
5260 LET B3=0
5280 PRINT
5300 PRINT "WOULD YOU LIKE THE SALES REPORT TO COVER A DATE"
5320 PRINT "OTHER THEN THAT OF TODAY (YES OR NO) ";
5340 INPUT Y$
5360 PRINT
5380 IF Y$="NO" THEN 5520
5400 IF Y$="N" THEN 5520
5420 PRINT
5440 PRINT "INPUT THE SALES DATE FOR THIS REPORT ";
5460 INPUT Q$
5480 PRINT
5500 GOTO 5560
5520 LET Q$=D$
5540 PRINT
5560 PRINT
5580 PRINT "          S A L E S     R E P O R T     F O R  ";Q$
5600 PRINT "                                        TODAY'S DATE ";D$
5620 PRINT
5640 PRINT "  ACC# ","  ITEM#"," QUANTITY "," SALES $"," CUST NAME"
5660 PRINT
5680 RESTORE
5700 FOR I=1 TO N1
5720 READ A(I),N$(I),R$(I),S$(I),U(I),U$(I),B(I)
5740 IF B(I)=0 THEN 5900
5760 FOR I1=1 TO B(I)
5780 READ E$(I),N(I),A$(I),S(I),P(I)
5800 IF Q$<>A$(I) THEN 5880
5820 PRINT A(I),N(I),S(I),S(I)*P(I);"    ";N$(I)
5840 B1=B1+S(I)
5860 B2=B2+S(I)*P(I)
5880 NEXT I1
5900 NEXT I
5920 IF B1<>0 THEN 5980
5940 PRINT "THERE WAS NO ACTIVITY FOR THAT DATE - TRY AGAIN ?"
5960 GOTO 6180
5980 PRINT
6000 PRINT
6020 PRINT
6040 PRINT "TOTAL PIECES SOLD THIS DATE   : ";B1
6060 T2=B2*S
6080 T2=INT(.5+100*T2)/100
6100 PRINT "TOTAL SALES TAX THIS DATE     : ";T2
6120 T2=T2+B2
6140 PRINT "TOTAL RECEIPTS THIS DATE      :";T2
```

Page Was Missing From Book

Page Was Missing From Book

```
8200 REM
8220 REM
8240 REM
8260 REM
8280 REM
8300 REM
8320 REM
8340 REM
8360 REM
8380 REM
8400 REM
8420 REM
8440 REM
8460 REM
8480 REM
8500 REM
8520 REM
8540 REM
8560 REM
8580 REM
8600 REM
8620 REM LEAVE THESE THREE LINES ALONE - PUT YOUR DATA AROUND THESE LINES.
8640 GOTO 10000
8660 REM THE REMAINING LINES ARE RESERVED FOR FUTURE EXPANSION.
10000 END
```

Billing Modifications

This program is written using elementary Basic statements for the most part. This was done to make it compatible with most computing systems on the market. By adding or modifying a few of the statements this program can be made to perform as well as a billing program costing thousands of dollars; just for rental.

The first modification would be to use PRINT USING statements in place of the PRINT statements for the report print outs. This statement, which may not be included in a 5K Basic, will force the print outs to be columnar regardless of the field size or data content. Under PRINT statements for example, when prices are printed the trailing zeros are suppressed, if this happens in a dollar column the numbers may be miss read; See the sample listing - Report #2 (Westaim Sales Co.). To resolve this problem substitute PRINT USING statements to force all dollar amounts into a columnar format for the report print outs. Lines 3860, 3960, 4060 and 4120 should be changed to PRINT USING statements in order to modify Report #2. For Report #3, the A/R report, change the PRINT statements in program lines 4980, 5100 and 5120 to PRINT USING statements. Report #4 should have the PRINT statements in lines 5820, 6100 and 6140 changed to PRINT USING statements and Report #5 should have the PRINT statements in lines 6700 and 6840 changed. Line 6700 for Report #5 also contains the field designator which should limit the printing in the last field, customer name, to avoid lap over problems. With these slight changes the print outs will for the most part conform to the format consistent with the standard forms being used. As a final word about formatting the print outs, have your TAB( ) statements set to absorb variations in form widths. These TAB( ) settings may have to be changed slightly to initially line up with the forms you are using. For the mailing labels, Report #1, the tab control is in line 2400. Line 3320 sets the number of spaces to initially skip to reach the first bill heading for Report #2 and line 3480 sets the number of spaces the name and address will be moved to the right on the bill heads. This name should appear through the window if used in window envelopes. Line 3620 adjusts the number of skips between the bill head and the first line of the bill body. Z7 in line 3720 sets the location of the statement date and Z3 in line 3820 designates the start printing position for the billing data. Line 4140 is used to skip down to the head of the next bill.

The only other modification that could enhance this program would be to remove the DATA statements and change the READ statements to READ # statements. This must be accompanies with a FILES statement and WRITE # statements. When completed this program will read your customer data from a Use File. Upon inclusion of WRITE # statements the program will update when applicable your customer data automatically. Some individuals prefer to store their data base in a Use File rather then DATA statements, however the way this program is written nothing is gained by doing this. If however, the PRINT USING, PRINT and READ # statements are merged then the program will execute in about 14K Bytes of memory even with 1000 customers which is a considerable savings in memory usage. As file manipulations are vastly different from system to system, to present this program in its compactest form would restrict its use to all but the particular computer it was written for, which would defeat the intentions of this Library.

THIS IS THE BILLING AND ACCOUNTS RECEIVEABLE PROGRAM

TYPE IN TODAY'S DATE, AS: 1-14/75 ?3/8/76

THE FOLLOWING IS A LIST OF REPORTS THAT THIS PROGRAM
WILL GENERATE. TO CHOOSE ONE TYPE IT'S NUMBER WHEN ASKED.

        1 - PRINT MAILING LABELS
        2 - PRINT UP BILLS
        3 - CUSTOMER A/R REPORT
        4 - SALES REPORT
        5 - LAST PURCHASE REPORT
        6 - STOP PROGRAM

WHICH REPORT WOULD YOU LIKE TO RUN: ?1

  THIS IS THE MAILING LIST ROUTINE

WHEN YOU HAVE THE LABELS IN PLACE TYPE A 7: ??

        37116
    WESTAIM SALES CO.
    327 W. CHERRY ST.
    MELROSE IL 37215

        45686
    REMINGTON CAN
    2314 SHERWOOD AVE.
    LAKESIDE TX 45217

        51376
    OMEGA IND. CO
    PO BOX 4045
    STANFORD NJ 91268

        78192
    DIGITRAN CO.
    3421 ST. LAWRENCE
    PASADENA CA 91105

END OF THE MAILING LIST *****

THE FOLLOWING IS A LIST OF REPORTS THAT THIS PROGRAM
WILL GENERATE. TO CHOOSE ONE TYPE IT'S NUMBER WHEN ASKED.

        1 - PRINT MAILING LABELS
        2 - PRINT UP BILLS
        3 - CUSTOMER A/R REPORT
        4 - SALES REPORT
        5 - LAST PURCHASE REPORT
        6 - STOP PROGRAM

WHICH REPORT WOULD YOU LIKE TO RUN: ?2

THIS GENERATES THE PRINTING OF THE BILLS
WHEN THEY ARE IN PLACE TYPE A 7 :?7

        37116
    WESTAIM SALES CO.
    327 W. CHERRY ST.
    MELROSE IL 37215

                                                        3/8/76

    556178   ADJ. PULLEY   25   11/10/75   28.5
    876512   WIRE MESH   50   12/11/75   4.25
    136928   ADJ. WRENCH   20   12/13/75   9.98
    221679   HAND SAW   14   1/7/76   7.98

TOTAL AMOUNT PAID   = $ 711.26
DATE OF LAST PAYMENT =   12/17/75
TOTAL PIECES SHIPPED =   109
TOTAL SALES TAX AMT. = $ 49.45
TOTAL AMOUNT DUE   =   ********** $ 574.51 **********

558

```
     45686
REMINGTON CAN
2314 SHERWOOD AVE.
LAKESIDE TX 45217


                                                              3/3/76

     106928    ADJ. WRENCH          2    1/16/76             9.98
     234561    PLASTIC ROD          1    1/29/76             4.75

TOTAL AMOUNT PAID     = $   21.76
DATE OF LAST PAYMENT  =   2/17/76
TOTAL PIECES SHIPPED  =        3
TOTAL SALES TAX AMT.  = $   .99
TOTAL AMOUNT DUE      =  **********   $ 3.94   **********




     78192
DIGITRAN CO.
3421 ST. LAWRENCE
PASADENA CA 91105


                                                              3/3/76

     723756    ELEC. BOX            6    1/13/76            29.66
     745336    FUSE BLOCK           6    1/13/76            15.95
     234561    PLASTIC ROD          5    2/3/76              4.75

TOTAL AMOUNT PAID     = $   213.5
DATE OF LAST PAYMENT  =   2/1/76
TOTAL PIECES SHIPPED  =       17
TOTAL SALES TAX AMT.  = $ 11.9
TOTAL AMOUNT DUE      =  **********   $   95.91   **********
```

```
      93216
SUPERIOR ELEC.
BOX 30
READER PA 16604


                                                                    2/13/76

      812763    GLOBE                 8    10/16/75               7.19
      915332    FILE                  7    11/23/75               1.98
      234561    PLASTIC ROD           9    12/2/76                4.75
      221679    SM.HAND SAW           1    1/21/76                7.98

TOTAL AMOUNT PAID      =  $  113.35
DATE OF LAST PAYMENT   =   2/6/76
TOTAL PIECES SHIPPED   =     25
TOTAL SALES TAX AMT.   = $ 4.98
TOTAL AMOUNT DUE       =   ***********  $   16.14   ***********
```

END OF THE BILL PRINTING SEQUENCE.

THE FOLLOWING IS A LIST OF REPORTS THAT THIS PROGRAM
WILL GENERATE. TO CHOOSE ONE TYPE IT'S NUMBER WHEN ASKED.

        1 - PRINT MAILING LABELS
        2 - PRINT UP BILLS
        3 - CUSTOMER A/R REPORT
        4 - SALES REPORT
        5 - LAST PURCHASE REPORT
        6 - STOP PROGRAM

WHICH REPORT WOULD YOU LIKE TO RUN: ?3

C U S T O M E R S   A / R   R E P O R T   3/8/76

ACC#	A/R $	PAYMENTS	LS DATE	CUST NAME
37116	574.51	711.26	12/17/75	WESTAIN SALES CO.
45686	3.94	21.76	2/17/76	REMINGTON CAN
78192	95.81	213.5	2/1/76	DIGITRAN CO.
93216	16.14	113.35	2/6/76	SUPERIOR ELEC.

TOTAL PAYMENTS   = $ 1059.87
TOTAL ACCOUNTS/REC. = $ 690.4

THE FOLLOWING IS A LIST OF REPORTS THAT THIS PROGRAM
WILL GENERATE. TO CHOOSE ONE TYPE IT'S NUMBER WHEN ASKED.

      1 - PRINT MAILING LABELS
      2 - PRINT UP BILLS
      3 - CUSTOMER A/R REPORT
      4 - SALES REPORT
      5 - LAST PURCHASE REPORT
      6 - STOP PROGRAM

WHICH REPORT WOULD YOU LIKE TO RUN: ?4

WOULD YOU LIKE THE SALES REPORT TO COVER A DATE
OTHER THEN THAT OF TODAY (YES OR NO) ?YES

INPUT THE SALES DATE FOR THIS REPORT ?1/13/76


```
          S A L E S      R E P O R T      F O R   1/13/76
                                          TODAYS DATE 3/8/76

     ACC#            ITEM#         QUANTITY        SALES $          CUST NAME

       78192          723756              6       177.96     DIGITRAN CO.
       78192          745336              6       95.7      DIGITRAN CO.



TOTAL PIECES SOLD THIS DATE   :      12
TOTAL SALES TAX THIS DATE     :      10.95
TOTAL RECEIPTS THIS DATE      :    284.61
```


WOULD YOU LIKE TO CHECK THE SALES FOR ANOTHER DATE :?YES

WOULD YOU LIKE THE SALES REPORT TO COVER A DATE
OTHER THEN THAT OF TODAY (YES OR NO) ?YES

INPUT THE SALES DATE FOR THIS REPORT ?1/21/76


```
          S A L E S      R E P O R T      F O R   1/21/76
                                          TODAYS DATE 3/8/76

     ACC#            ITEM#         QUANTITY        SALES $          CUST NAME

       93216          221679              1       7.98     SUPERIOR ELEC.



TOTAL PIECES SOLD THIS DATE   :       1
TOTAL SALES TAX THIS DATE     :      .32
TOTAL RECEIPTS THIS DATE      :    8.3
```

WHICH REPORT WOULD YOU LIKE TO RUN: ?5

L A S T   P U R C H A S E   R E P O R T   3/9/76

ACC#	ITEM#	QUANTITY	DATE	CUST. NAME
37116	221679	14	1/7/76	WESTAIM SALES CO.
45686	234561	1	1/29/76	PEMINGTON CAN
78192	234561	5	2/3/76	DIGITRAN CO.
93216	221679	1	1/21/76	SUPERIOR ELEC.

TOTAL LAST SALES : $ 154.13

THE FOLLOWING IS A LIST OF REPORTS THAT THIS PROGRAM
WILL GENERATE. TO CHOOSE ONE TYPE IT'S NUMBER WHEN ASKED.

        1 - PRINT MAILING LABELS
        2 - PRINT UP BILLS
        3 - CUSTOMER A/R REPORT
        4 - SALES REPORT
        5 - LAST PURCHASE REPORT
        6 - STOP PROGRAM

WHICH REPORT WOULD YOU LIKE TO RUN: ?6

INVENTORY:

DESCRIPTION

This program is designed to keep track of all inventory data. The program does not use external data files for data handling as all of the data is contained within the program itself. Items are categorized according to their item #, class code, location, vendor code, etc. Data fields are provided for item description, item cost, selling price, etc. The reports generated by this program include: (1) Activity Report, (2) Minimum Quantity Search, (3) Inventory List, (4) Inventory List by Class and (5) Inventory List by Vendor.

USERS

Anyone who wishes or has need to categorize the items in their possession could use this program as well as the small businessman with supplies and stock records to maintain. This would include housewives with kitchen inventories, hobbyist with equipment and parts inventories, do-it-yourselfers with tools and hdwr., and of course the small businessman or company.

INSTRUCTIONS

The data must be updated in the data statements before the program is run. The data is entered and handled in a similar manner as it is in the Billing program. This program is self documented and contains a full set of instructions for data handling. List the source code for detailed information.

LIMITATIONS

This program requires 11K Bytes of memory for source code storage. The amount of memory required for program execution is a function of the size of the DIM statements in lines 1000 and 1020. It is presently set for 100 classes and 100 subclasses. That amounts to a total inventory of 10,000 different types of items. With the size set to 100 the program will require 24K Bytes of available memory for execution. A sample run of this program is presented at the end of this program.
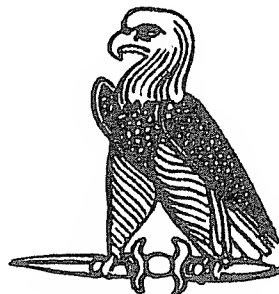
```
100 REM THIS IS THE INVENTORY PROGRAM
120 REM FOR FULL NOTIFICATIONS LIST THE PROGRAM
140 REM THIS PROGRAM IS WRITTEN IN BASIC
160 REM IT DOES NOT USE EXTERNAL USE FILES
180 REM ALL UPDATING IS TO BE DONE BEFORE EXECUTION
200 REM CREATED BY SCIENTIFIC RESEARCH INST.
220 REM ALL INVENTORY DATA IS CONTAINED IN THE DATA STATEMENTS
240 REM EACH TIME THE DATA IS UPDATED TO INCORPORATE CHANGES
260 REM IN INVENTORY STATUS A NEW COPY OF THE PROGRAM SHOULD
280 REM BE MADE. EITHER ON TAPE OR PAPER THEN
300 REM EACH TIME THE PROGRAM IS RUN THE NEW COPY
320 REM WILL BE USED TO INPUT THE PROGRAM TO
340 REM THE COMPUTER. THE PROGRAM IS SELF PROMPTING
360 REM AND WILL ASK ALL NECESSARY QUESTIONS.
380 REM THE SUBSCRIPT (100) USED IN THE DIM
400 REM STATEMENTS IS THE MAXIMUM NUMBER OF
420 REM INVENTORY ITEMS THIS PROGRAM IS SET UP TO HANDLE
440 REM TO INCREASE THIS NUMBER TO ANY OTHER NUMBER
460 REM SIMPLY CHANGE THIS NUMBER TO THE DESIRED INVENTORY
480 REM LEVEL IN EACH OF THE DIM STATEMENTS INITIALLY
500 REM BEFORE THE DATA IS INPUT TO THE DATA STATMENTS.
520 REM THE DATA STATEMENTS ARE USED TO UPDATE THE
540 REM INVENTORY EVERY PERIOD. WHATEVER THIS PERIOD HAPPENS TO
560 REM BE. THE DATA STATEMENTS ARE ARRANGED IN TWO (2)
580 REM LINES FOR EASE IN UPDATING. THE FIRST LINE
600 REM CONTAINS RELATIVELY UNCHANGING DATA - LIKE
620 REM THE ITEM#, CLASS#,VENDOR#,DESCRIPTION,COST$,LOCATION#
640 REM THE SECOND LINE WILL CONTAIN THE VARIABLE DATA LIKE
660 REM THE #ON HAND,#SOLD THIS PERIOD,DATE THIS PERIOD,SELLING PR$.
680 REM THE LAST ENTRY - DATE - IS ONLY USED TO INDICATE
700 REM THE LATEST SALES ACTIVITY FOR THE PARTICULAR ITEM.
720 REM THE FOLLOWING VARIABLES ARE USED IN THIS PROGRAM
740 REM N=TOTAL # OF INVENTORY ITEMS
760 REM N(I)=ITEM #
780 REM C(I)=CLASS OR SUBITEM #
800 REM V(I)=VENDOR #
820 REM E$(I)=DESCRIPTION (LESS THEN 20 CHARACTERS PLEASE)
840 REM O(I)=UNIT COST $
860 REM L(I)=LOCATION # DESIGNATES WHERE ITEM IS STORED.
880 REM LINE NUMBER TWO OF THE DATA STATEMENT NOW HAS
900 REM H(I)=QUANTITY ON HAND
920 REM S(I)=NUMBER OF SALES DURING THE LAST INVENTORY PERIOD
940 REM A$(I)=DATE OF THIS PERIOD - BE CONSISTENT EX: 12/18/75
960 REM P(I)=SELLING PRICE IN $
980 REM
1000 DIM N(100),C(100),V(100),E$(100),O(100),L(100),H(100),S(100),A$(100)
1020 DIM P(100)
1040 REM CHANGE THE ABOVE SUBSCRIPTS TO EQUAL THE NUMBER OF YOUR
1060 REM TOTAL INVENTORY ITEMS.
1080 REM *********========================
1100 N=10
```

```
1120 REM CHANGE THE VALUE OF N TO EQUAL YOUR TOTAL NUMBER OF ITEMS.
1140 REM *********:::::::::::::::::::::::::::::::::
1160 FOR I=1 TO N
1180 READ H(I),C(I),V(I),E$(I),O(I),L(I)
1200 READ H(I),S(I),A$(I),P(I)
1220 NEXT I
1240 REM THE FOLLOWING DATA STATEMENTS MUST BE CHANGED TO
1260 REM REFLECT YOU TRUE INVENTORY ITEMS AND CODES
1280 REM ALL OF THE CODES YOU DON'T USE MUST BE DEFINED
1300 REM SO PUT A 0 IN EACH ITEM YOU ARE NOT USING - EX: CLASS#
1320 REM THE DATA FORMAT FOR THE FIRST DATA LINE OF AN ITEM IS AS FOLLOWS:
1340 REM DATA ITEM#,CLASS#,VENDOR#,ITEM DESCRIPTION,UNIT COST$,LOCATION CODE#
1360 REM THE FORMAT FOR THE SECOND DATA LINE OF AN ITEM IS AS FOLLOWS:
1380 REM DATA #ON HAND,#SOLD THIS PERIOD,DATE OF THIS PERIOD,SELLING PR$
1400 REM *********--------------------------------------------
1420 DATA 136928,13,1673,ADJ. WRENCH,7.13,189
1440 DATA 9,0,"12/17/75",9.98
1460 DATA 221679,9,1673,SM. HAND SAW,5.17,173
1480 DATA 4,1,"2/23/76",7.98
1500 DATA 234561,0,96,PLASTIC ROD,2.18,27
1520 DATA 112,53,"1/19/76",4.75
1540 DATA 556178,2,873,ADJ. PULLEY,22.19,117
1560 DATA 3,0,"12/17/75",28.50
1580 DATA 723756,73,27,ELECT.BOX,19.56,354
1600 DATA 6,1,"2/23/76",29.66
1620 DATA 745336,13,27,FUSE BLOCK,12.65,63
1640 DATA 7,2,"12/17/75",15.95
1660 DATA 812763,2,1673,GLOBE,5.88,112
1680 DATA 5,2,"2/23/76",7.49
1700 DATA 876512,2,873,WIRE MESH,3.18,45
1720 DATA 7,3,"1/19/76",4.25
1740 DATA 915332,2,1673,FILE,1.32,97
1760 DATA 7,3,"2/23/76",1.98
1780 DATA 973328,0,27,CAN COVER,.73,21
1800 DATA 17,5,"12/17/75",.99
1820 REM
1840 REM
1860 REM
1880 REM
1900 REM
1920 REM
1940 REM
1960 REM
1980 REM
2000 REM
2020 REM
2040 REM
2060 REM
2080 REM
2100 REM
2120 REM
```

```
2140 REM
2160 REM
2180 REM
2200 REM
2220 REM THESE LINES ARE RESERVED FOR YOUR INVENTORY ITEMS.
2240 REM INITIALLY TYPE IN THE DATA IN ITEM # SEQUENCE IF POSSIBLE
2260 REM THIS WILL GREATLY SPEED UP THE FUTURE UPDATING THAT WILL
2280 REM BE REQUIRED EVERY INVENTORY PERIOD, WHATEVER IT IS.
2300 REM THE DATA DOES NOT NEED TO BE UPDATED TO BE RUN BUT IT
2320 REM IS DESIRABLE TO UPDATE THE INVENTORY ON A PERIODIC BASIS.
2340 REM IT WOULD BE A GOOD IDEA TO SKIP AT LEAST 3 LINE NUMBERS
2360 REM BETWEEN EACH DATA ENTRY. YOU WILL NOTICE THAT THE
2380 REM LINE NUMBERS ARE NOT INCREMENTED BY 1 BUT INSTEAD BY 20
2400 REM THIS MEANS THAT THERE ARE 20 POSSIBLE DATA LINES BETWEEN
2420 REM EACH ABOVE REM STATEMENT.
2440 REM
2460 PRINT
2480 PRINT
2500 PRINT "   THIS IS THE INVENTORY ACCOUNTING PROGRAM "
2520 PRINT
2540 PRINT
2560 PRINT
2580 PRINT "TYPE IN THE DATE, AS: 1/14/75 ";
2600 INPUT D$
2620 PRINT
2640 FOR I=1 TO 7
2660 PRINT
2680 NEXT I
2700 PRINT
2720 PRINT
2740 PRINT
2760 PRINT "THE FOLLOWING IS A LIST OF THE REPORTS THAT ARE AVAILABLE"
2780 PRINT "TO CHOOSE ONE TYPE IT'S NUMBER TO THE PROGRAM PROMPT."
2800 PRINT
2820 PRINT "      1 = ACTIVITY REPORT "
2840 PRINT "      2 = MINIMUM QUANTITY SEARCH "
2860 PRINT "      3 = INVENTORY LIST "
2880 PRINT "      4 = INVENTORY LIST BY CLASS "
2900 PRINT "      5 = INVENTORY LIST BY VENDOR "
2920 PRINT "      6 = STOP PROGRAM "
2940 PRINT
2960 PRINT "WHICH REPORT WOULD YOU LIKE TO RUN:";
2980 INPUT R
3000 PRINT
3020 IF R=6 THEN 7600
3040 IF R=1 THEN 3180
3060 IF R=2 THEN 4540
3080 IF R=3 THEN 5180
3100 IF R=4 THEN 5860
3120 IF R=5 THEN 6800
3140 PRINT
```

```
3160 GOTO 2820
3180 REM THIS SUB GENERATES THE ACTIVITY REPORT  # 1
3200 LET A=5
3220 FOR I=1 TO A
3240 PRINT
3260 NEXT I
3280 PRINT "WOULD YOU LIKE YOUR ACTIVITY REPORT TO COVER A DATE"
3300 PRINT "OTHER THEN THAT OF TODAY (YES OR NO)";
3320 B1=0
3340 B2=0
3360 B3=0
3380 A1=0
3400 A2=0
3420 A3=0
3440 INPUT Y$
3460 PRINT
3480 IF Y$="NO" THEN 3740
3500 IF Y$="N" THEN 3740
3520 PRINT
3540 B1=0
3560 B2=0
3580 B3=0
3600 PRINT "INPUT THE PERIOD DATE FOR THIS REPORT ";
3620 A1=0
3640 A2=0
3660 A3=0
3680 INPUT O$
3700 PRINT
3720 GOTO 3780
3740 O$=D$
3760 PRINT
3780 PRINT
3800 PRINT "   A C T I V I T Y   R E P O R T   F O R ";O$
3820 PRINT
3840 PRINT "                                    TODAY'S DATE ";T$
3860 PRINT
3880 PRINT " ITEM# "," DESCRIPTION "," #ON HAND "," $SALES "," $PROFIT"
3900 PRINT
3920 A7=0
3940 FOR I=1 TO N
3960 A8=S(I)*(P(I)-O(I))
3980 IF A$(I)=O$ THEN 4040
4000 NEXT I
4020 GOTO 4180
4040 PRINT H(I),E$(I),H(I),S(I)*P(I),A8
4060 A3=A3+A8
4080 A2=A2+S(I)*P(I)
4100 A1=A1+1
4120 B1=B1+S(I)
4140 A7=1
4160 GOTO 4000
```

```
4180 IF A7=1 THEN 4240
4200 PRINT
4220 PRINT "THERE WAS NO ACTIVITY DURING THAT PERIOD "
4240 PRINT
4260 PRINT "TOTAL PIECES SOLD   = ";B1
4280 PRINT "TOTAL ITEMS         = ";A1
4300 PRINT "TOTAL DOLLAR SALES  = $";A2
4320 PRINT "TOTAL DOLLAR PROFIT = $";A3
4340 PRINT
4360 PRINT
4380 PRINT
4400 PRINT
4420 PRINT "WOULD YOU LIKE TO CHECK THE ACTIVITY FOR ANOTHER PERIOD ";
4440 INPUT Y$
4460 PRINT
4480 IF Y$="YES" THEN 3520
4500 IF Y$="Y" THEN 3520
4520 GOTO 2620
4540 REM THIS SUB GENERATES A SEARCH FOR MINIMUM QUANTITIES
4560 REM THIS IS REPORT #2
4580 LET A=5
4600 FOR I=1 TO A
4620 PRINT
4640 NEXT I
4660 PRINT "    M I N I M U M    Q U A N T I T Y    S E A R C H ";D$
4680 PRINT
4700 PRINT
4720 PRINT "TYPE IN THE MINIMUM QUANTITY LEVEL THAT YOU ARE INTERESTED"
4740 PRINT "IN. FOR EXAMPLE TO CHECK FOR 0 STOCK ITEMS TYPE A - 1"
4760 PRINT "TO CHECK FOR ITEMS WITH QUANTITIES LESS THEN 5 TYPE A - 5"
4780 PRINT "TYPE A 0 IF YOU WANT ANOTHER REPORT INSTEAD. INPUT QUANTITY: ";
4800 INPUT A5
4820 IF A5=0 THEN 2620
4840 IF A5<0 THEN 2620
4860 PRINT
4880 PRINT
4900 PRINT "   MINIMUM QUANTITY ON HAND SEARCH REPORT   ";D$
4920 PRINT
4940 PRINT
4960 PRINT " ITEM# "," DESCRIPTION "," #ON HAND "," #SOLD "," LOCAT#"
4980 PRINT
5000 FOR I=1 TO N
5020 IF H(I)<A5 THEN 5100
5040 IF H(I)=A5 THEN 5100
5060 NEXT I
5080 GOTO 5140
5100 PRINT N(I),E$(I),H(I),S(I),L(I)
5120 GOTO 5060
5140 PRINT
5160 GOTO 2620
5180 REM THIS IS THE INVENTORY LIST SUB #3
```

```
5200 LET A=5
5220 FOR I=1 TO A
5240 PRINT
5260 NEXT I
5280 PRINT "        L I S T     O F     I N V E N T O R Y   ";D$
5300 PRINT
5320 PRINT
5340 PRINT TAB(3);
5360 PRINT "  ITEM # "," CLASS # "," VENDOR #"," COST $"," DESCRIPTION"
5380 PRINT
5400 PRINT " LOCAT# "," QUAN# "," #SOLD "," SEL PR$ ","LST DATE"
5420 PRINT
5440 PRINT
5460 A1=0
5480 A2=0
5500 A3=0
5520 A4=0
5540 FOR I=1 TO N
5560 PRINT N(I),C(I),V(I),O(I),E$(I)
5580 PRINT L(I),H(I),S(I),P(I),A$(I)
5600 A1=A1+O(I)*H(I)
5620 A2=A2+P(I)*S(I)
5640 A3=A3+H(I)
5660 PRINT
5680 NEXT I
5700 PRINT
5720 PRINT
5740 PRINT "PRESENT INVENTORY COST  = $";A1
5760 PRINT "TOTAL NUMBER OF PIECES  =  ";A3
5780 PRINT "TOTAL LST SALES PERIODS = $";A2
5800 PRINT "TOTAL NUMBER OF INVENTORY ITEM ON RECORD = ";N
5820 PRINT
5840 GOTO 2620
5860 REM THIS SUB GENERATES AN INVENTORY LISTING BY CLASS  #4
5880 A=5
5900 FOR I=1 TO A
5920 PRINT
5940 NEXT I
5960 PRINT "THIS REPORT GENERATES A LIST BY CLASS CODE FROM THE ITEMS"
5980 PRINT "PRESENTLY ON YOUR INVENTORY RECORD."
6000 PRINT "TYPE IN THE CLASS CODE YOU WANT SEARCHED:";
6020 INPUT A5
6040 B1=0
6060 B2=0
6080 B3=0
6100 PRINT
6120 PRINT
6140 PRINT
6160 PRINT "         C L A S S     C O D E     L I S T ";D$
6180 PRINT
6200 PRINT
```

```
6220 PRINT " CLASS# ","  ITEM# "," DESCRIP. "," #ON HAND"," VENDOR#"
6240 PRINT
6260 A7=A
6280 FOR I=1 TO N
6300 IF C(I)=A5 THEN 6360
6320 NEXT I
6340 GOTO 6480
6360 PRINT C(I),N(I),E$(I),H(I),U(I)
6380 B1=B1+H(I)*O(I)
6400 B2=B2+1
6420 B3=B3+H(I)
6440 A7=1
6460 GOTO 6320
6480 IF A7=1 THEN 6540
6500 PRINT "CAN'T FIND THAT CLASS CODE IN THE INVENTORY LIST RECHECK CODE#"
6520 PRINT
6540 PRINT
6560 PRINT "TOTAL CLASS COST    =  $";B1
6580 PRINT "TOTAL NUMBER OF ITEMS  =";B2
6600 PRINT "TOTAL NUMBER OF PIECES =";B3
6620 PRINT
6640 PRINT "WOULD YOU LIKE TO CHECK ANOTHER CLASS CODE (YES OR NO) ";
6660 INPUT Y$
6680 PRINT
6700 PRINT
6720 IF Y$="YES" THEN 6000
6740 IF Y$="Y" THEN 6000
6760 PRINT
6780 GOTO 2620
6800 REM THIS SUB GENERATES THE VENDOR CODE LIST  # 5
6820 LET A=5
6840 FOR I=1 TO A
6860 PRINT
6880 NEXT I
6900 PRINT "TYPE IN THE VENDOR CODE YOU WOULD LIKE TO SEARCH ";
6920 INPUT A5
6940 PRINT
6960 PRINT
6980 PRINT "      V E N D O R   C O D E   L I S T  ";A$
7000 PRINT
7020 PRINT
7040 PRINT " VENDOR# "," ITEM# "," CLASS "," #ON HAND"," DESC."
7060 PRINT
7080 A1=0
7100 A7=0
7120 FOR I=1 TO N
7140 IF U(I)=A5 THEN 7200
7160 NEXT I
7180 GOTO 7280
7200 PRINT U(I),N(I),C(I),H(I),E$(I)
7220 A7=1
```

```
7240 A1=A1+1
7260 GOTO 7160
7280 IF A7=1 THEN 7340
7300 PRINT
7320 PRINT "COULDN'T FIND THAT VENDOR IN THE INVENTORY LIST - TRY AGAIN"
7340 PRINT
7360 PRINT "TOTAL NUMBER OF ITEMS = ";A1
7380 PRINT
7400 PRINT
7420 PRINT
7440 PRINT
7460 PRINT "WOULD YOU LIKE TO CHECK ANOTHER VENDOR ";
7480 INPUT Y$
7500 PRINT
7520 PRINT
7540 IF Y$="YES" THEN 6900
7560 IF Y$="Y" THEN 6900
7580 GOTO 2620
7600 END
```
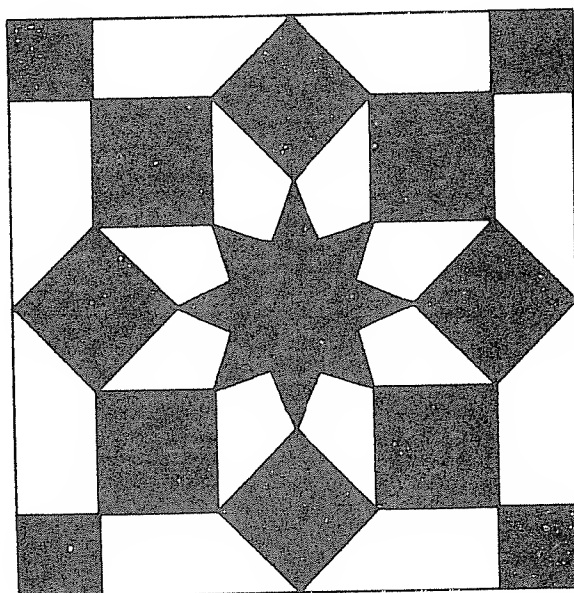
Inventory Modifications

This program has been written using no FILES statements or PRINT USING statements. This was done to assure maximum compatibility with the various Basic compilers currently on the market. The operation of this program can be refined by substituting PRINT USING statements in place of some of the PRINT statements controlling report print outs. If your Basic compiler or interpreter doesn't incorporate the Print Using statements this program without any modifications will function quite normally without error as currently written, however some of the dollar amounts may be printed right hand justified with the trailing decimal zeros suppressed; See the sample listing of the Activity Report, Report #1.

The PRINT USING statements could be substituted in reports #1, 3 and 4, for increased readability. For Report #1 amend lines 4040, 4300 and 4320 by inserting PRINT USING statements for the PRINT statements. For Report #3 lines 5560, 5580, 5740 and 5780 should have their PRINT statements converted to PRINT USING statements. In Report #4, the Class Code listing, line 6560 should be changed to a PRINT USING statement. The PRINT USING statements will each require different field designators in the control statement lines to assure printing conformity.

The DATA statements may be removed from this program if the data is to be read in from a Use File. If this is done the READ statements will have to be changed to READ # statements and a FILES statement will have to be inserted around program line 1000. The inclusion of an external Use File can greatly reduce the amount of on line memory required for program execution, if the program is written with this in mind. External Use Files were omitted in this version of the program due to compatibility variations between systems. If such a modification is deemed advantageous the READ # and PRINT and PRINT USING statements should be merged and the tables removed to conserve memory requirements.

THIS IS THE INVENTORY ACCOUNTING PROGRAM

TYPE IN THE DATE, AS: 1/14/75 ?3/24/76

THE FOLLOWING IS A LIST OF THE REPORTS THAT ARE AVAILABLE
TO CHOOSE ONE TYPE IT'S NUMBER TO THE PROGRAM PROMPT.

```
        1 = ACTIVITY REPORT
        2 = MINIMUM QUANTITY SEARCH
        3 = INVENTORY LIST
        4 = INVENTORY LIST BY CLASS
        5 = INVENTORY LIST BY VENDOR
        6 = STOP PROGRAM
```

WHICH REPORT WOULD YOU LIKE TO RUN:?1

WOULD YOU LIKE YOUR ACTIVITY REPORT TO COVER A DATE
OTHER THEN THAT OF TODAY (YES OR NO)?YES

INPUT THE PERIOD DATE FOR THIS REPORT ?12/17/75

     A C T I V I T Y   R E P O R T   F O R 12/17/75

                              TODAY'S DATE 3/24/76

   ITEM#   DESCRIPTION   #ON HAND $SALES $PROFIT

      106928   ADJ. WRENCH   9   0
      556179   ADJ. PULLEY   8   0
      745336   FUSE BLOCK   7   31.9   6.0
      973399   FAN COVER   17   4.95   1.9

TOTAL PIECES SOLD   =   7
TOTAL ITEMS   =   4
TOTAL DOLLAR SALES = $ 36.85
TOTAL DOLLAR PROFIT = $ 7.9

WOULD YOU LIKE TO CHECK THE ACTIVITY FOR ANOTHER PERIOD ?YES

INPUT THE PERIOD DATE FOR THIS REPORT ?12/24/75

     A C T I V I T Y   R E P O R T   F O R 12/24/75

                              TODAY'S DATE 3/24/76

   ITEM#   DESCRIPTION   #ON HAND $SALES $PROFIT

THERE WAS NO ACTIVITY DURING THAT PERIOD

TOTAL PIECES SOLD   =   0
TOTAL ITEMS   =   0
TOTAL DOLLAR SALES = $ 0
TOTAL DOLLAR PROFIT = $ 0

WOULD YOU LIKE TO CHECK THE ACTIVITY FOR ANOTHER PERIOD ?NO

THE FOLLOWING IS A LIST OF THE REPORTS THAT ARE AVAILABLE
TO CHOOSE ONE TYPE IT'S NUMBER TO THE PROGRAM PROMPT.

```
    1 = ACTIVITY REPORT
    2 = MINIMUM QUANTITY SEARCH
    3 = INVENTORY LIST
    4 = INVENTORY LIST BY CLASS
    5 = INVENTORY LIST BY VENDOR
    6 = STOP PROGRAM
```

WHICH REPORT WOULD YOU LIKE TO RUN:?2

M I N I M U M   Q U A N T I T Y   S E A R C H 3/24/76

TYPE IN THE MINIMUM QUANTITY LEVEL THAT YOU ARE INTERESTED
IN. FOR EXAMPLE TO CHECK FOR 0 STOCK ITEMS TYPE A - 1
TO CHECK FOR ITEMS WITH QUANTITIES LESS THEN 5 TYPE A - 5
TYPE A 0 IF YOU WANT ANOTHER REPORT INSTEAD. INPUT QUANTITY: ?10

MINIMUM QUANTITY ON HAND SEARCH REPORT   3/24/76

ITEM#	DESCRIPTION	#ON HAND	#SOLD	LOCAT#
136928	ADJ. WRENCH	9	0	199
221679	SM. HAND SAW	4	1	173
556178	ADJ. PULLEY	3	0	117
723756	ELECT.BOX	6	1	354
745336	FUSE BLOCK	7	2	43
812763	GLOBE	5	2	112
876512	WIRE MESH	7	3	45
915332	FILE	7	3	97

WHICH REPORT WOULD YOU LIKE TO RUN:?3

L I S T   O F   I N V E N T O R Y   3/24/76

ITEM #	CLASS #	VENDOR #	COST $	DESCRIPTION
LOCAT#	QUAN#	#SOLD	SEL PR$	LST DATE
136928	13	1673	7.13	ADJ. WRENCH
189	9	0	9.98	12/17/75
221679	9	1673	5.17	SM. HAND SAW
173	4	1	7.98	2/23/76
234561	8	96	2.18	PLASTIC ROD
27	112	43	4.75	1/19/76
556178	8	873	22.19	ADJ. PULLEY
117	3	0	28.5	12/17/75
723756	73	27	19.56	ELECT. BOX
354	6	1	29.66	2/23/76
745336	13	27	12.35	FUSE BLOCK
63	7	2	15.95	12/17/75
812763	2	1673	5.38	GLOBE
112	5	2	7.49	2/23/76
876512	8	873	3.13	WIRE MESH
45	7	3	4.25	1/19/76
915332	8	1673	1.32	FILE
97	7	1	1.90	2/23/76
973328	8	27	.73	TIN COVER
21	17	5	.94	12/17/75

PRESENT INVENTORY COST   = $ 674.8
TOTAL NUMBER OF PIECES   =   177
TOTAL LST SALES PERIODS = $ 359.91
TOTAL NUMBER OF INVENTORY ITEM ON RECORD =   10

THE FOLLOWING IS A LIST OF THE REPORTS THAT ARE AVAILABLE
TO CHOOSE ONE TYPE IT'S NUMBER TO THE PROGRAM PROMPT.

```
    1 = ACTIVITY REPORT
    2 = MINIMUM QUANTITY SEARCH
    3 = INVENTORY LIST
    4 = INVENTORY LIST BY CLASS
    5 = INVENTORY LIST BY VENDOR
    6 = STOP PROGRAM
```

WHICH REPORT WOULD YOU LIKE TO RUN:?4

THIS REPORT GENERATES A LIST BY CLASS CODE FROM THE ITEMS
PRESENTLY ON YOUR INVENTORY RECORD.
TYPE IN THE CLASS CODE YOU WANT SEARCHED:?2

C L A S S   C O D E   L I S T   3/24/76

CLASS#	ITEM#	DESCRIP.	#ON HAND	VENDOR#
2	556179	ADJ. PULLEY	3	873
2	812763	GLOBE	5	1673
2	876512	WIRE MESH	7	173
2	915332	FILE	7	1673

```
TOTAL CLASS COST    = $  127.47
TOTAL NUMBER OF ITEMS =    4
TOTAL NUMBER OF PIECES =   22
```

WOULD YOU LIKE TO CHECK ANOTHER CLASS CODE (YES OR NO) ?NO

THE FOLLOWING IS A LIST OF THE REPORTS THAT ARE AVAILABLE
TO CHOOSE ONE TYPE IT'S NUMBER TO THE PROGRAM PROMPT.

```
        1 = ACTIVITY REPORT
        2 = MINIMUM QUANTITY SEARCH
        3 = INVENTORY LIST
        4 = INVENTORY LIST BY CLASS
        5 = INVENTORY LIST BY VENDOR
        6 = STOP PROGRAM
```

WHICH REPORT WOULD YOU LIKE TO RUN:?5

TYPE IN THE VENDOR CODE YOU WOULD LIKE TO SEARCH ?27

V E N D O R   C O D E   L I S T   3/24/76

VENDOR#	ITEM#	CLASS	#ON HAND	DESC.
27	723756	73	6	ELECT. BOX
27	745336	13	7	FUSE BLOCK
27	973328	0	17	LAY COVER

TOTAL NUMBER OF ITEMS =   3

WOULD YOU LIKE TO CHECK ANOTHER VENDOR ?YES

TYPE IN THE VENDOR CODE YOU WOULD LIKE TO SEARCH ?817

V E N D O R   C O D E   L I S T   3/24/76

VENDOR#	ITEM#	CLASS	#ON HAND	DESC.

COULDN'T FIND THAT VENDOR IN THE INVENTORY LIST - TRY AGAIN

TOTAL NUMBER OF ITEMS =   0

PAYROLL:

DESCRIPTION

This program calculates and compiles a payroll register for all of your employees. All employee data is contained within the program so that external data files need not be used. The program computes the deductions for FICA and Federal & State income tax withholdings, permits deductions for employee insurance, calculates the employers state and federal unemployment insurance tax and has space provided for an additional deduction calculation (Union dues, emp. savings, loan repymts, etc.) should such space be needed or required. Four different print outs are generated by this program: (1) printing of paychecks, (2) Payroll Register; in a tabulated format, (3) employee data record and (4) a summerized tax record for the employer.

USERS

This program is especially designed for small companies having a standardized work week and whose employees are paid by the hour rather then salaried.

INSTRUCTIONS

Before the program is run all employee data must initially be entered into the program. The program is well documented and should be listed for full details. The subroutines for the Federal taxes; line 1475, the Federal deduction schedule; line 1550, tax rate due; line 1585 and the State withholding; line 1680, should all be completed for your particular requirements before the program is initially run.

LIMITATIONS

This program is set for a maximum of 50 employees. This can be adjusted to accommodate other numbers of employees by changing the DIM statements in lines 155 and 160. The source code requires 9K Bytes of memory for storage and 15K Bytes of memory for execution, with 50 employees. A sample run of this program follows the source listing. The data generating the examples is contained within the program and should be removed before entering your data.
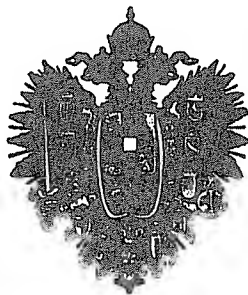
```
10 REM THIS IS        PAYROLL WITH DEDUCTIONS
15 REM FOR FULL MODIFICATIONS LIST THE PROGRAM
20 REM THIPROGRAM IS WRITTEN IN BASIC
25 REM IT DOES NOT USE EXTERNAL USE FILES
30 REM ALL UPDATING IS TO BE DONE BEFORE EXECUTION
35 REM CREATED BY SCIENTIFIC RESEARCH INST.
40 REM ALL EMPLOYEE DATA IS CONTAINED IN THE DATA STATEMENTS
45 REM EACH TIME THE DATA IS UPDATED TO INCORPORATE CHANGES
50 REM IN EMPLOYEE STATUS A NEW COPY OF THE PROGRAM SHOULD
55 REM BE MADE, EITHER ON TAPE OR PAPER THEN
60 REM EACH TIME THE PROGRAM IS RUN THE NEW COPY
65 REM WILL BE USED TO INPUT THE PROGRAM TO
70 REM THE COMPUTER, THE PROGRAM IS SELF PROMPTING
75 REM AND WILL ASK ALL NECESSARY QUESTIONS.
80 REM THE SUBSCRIPT (50) USED IN THE DIMENSION
85 REM STATEMENTS IS THE MAXIMUM NUMBER OF
90 REM EMPLOYEES. THIS NUMBER SHOULD BE EQUAL
95 REM TO THE NUMBER OF EMPLOYEES AND NO MORE
100 REM FOR EFFICIENT UTILIZATION OF MEMORY.
105 REM THE FOLLOWING VARIABLES ARE USED IN THIS PROGRAM
110 REM N=TOTAL # OF EMPLOYEES
115 REM E(I)=EMPLOYEE NUMBER
120 REM E$(I)=EMPLOYEE NAME
125 REM R(I)=HOURLY RATE OF PAY
130 REM D(I)=# OF DEPENDENTS (FOR TAX PURPOSES)
135 REM F(I)=FIXED $XXX.XX DEDUCTION (INS., DUES, ETC.)
140 REM C(I)=MISC. DEDUCTION (IF NONE MAKE 0)
145 REM AND C(I)=XXX.XX IN DOLLARS
150 REM H(I)=# OF HOURS WORKED
155 DIM E$(50),E(50),R(50),D(50),F(50),C(50)
160 DIM H(50)
165 REM CHANGE THE ABOVE SUBSCRIPTS FOR YOUR # OF EMPLOYEES
170 REM ********----------------------------------------
175 LET N=3
180 REM ???????------------------------------------------
185 REM THE ABOVE N MUST BE SET = TO YOUR # OF EMPLOYEES.
190 FOR I=1 TO N
195 READ E(I), E$(I), R(I), D(I), F(I), C(I)
200 NEXT I
205 REM THE FOLLOWING DATA STATEMENTS MUST BE CHANGED TO
210 REM REFLECT YOUR EMPLOYEE RECORDS.
215 REM THE FORMAT FOR THE DATA STATEMENTS ARE AS FOLLOWS
220 REM DATA EMP.#, NAME, RATE, DEPENDENTS, DEDUCTION, MISC. DED.
225 REM ********----------------------------------------
230 DATA 3721,JOHN M. DOYLE, 6.53,2,7.5,0
235 DATA 6219,G. R. CRAMER,3.1788,3,5.92,0
240 DATA 1872,PHIL JOHNSON,4.38,1,6.15,0
245 REM
250 REM
```

```
255 REM
260 REM
265 REM
270 REM
275 REM
280 REM
285 REM
290 REM
295 REM
300 REM
305 REM
310 REM
315 REM
320 REM
325 REM THESE LINES ARE RESERVED FOR FUTURE EMPLOYEE EXPANSIONS
330 PRINT
335 PRINT
340 PRINT "TYPE IN THE NUMBER OF HOURS IN THIS PAY PERIOD:";
345 INPUT H
350 FOR I2=1 TO N
355 LET H(I2)=H
360 NEXT I2
365 PRINT
370 PRINT "ENTER THE DATE (EX: 11/23/75 ):";
375 INPUT D$
380 PRINT
385 PRINT "DID ALL EMPLOYEES WORK THE FULL # OF HOURS (YES OR NO):";
390 INPUT A$
395 IF A$="YES" THEN 500
400 IF A$="Y" THEN 500
405 PRINT
410 PRINT "ENTER THE NUMBER OF EMPLOYEES THAT DIDN'T WORK THE FULL"
415 PRINT "NUMBER OF HOURS FOR THIS PAY PERIOD:";
420 INPUT I1
425 PRINT
430 PRINT "TYPE IN THE EMPLOYEE #, THEN # OF HOURS WORKED: AS ASKED"
435 FOR I2=1 TO I1
440 PRINT "INPUT-EMP#, # HOURS WK:";
445 INPUT G, J
450 FOR I=1 TO N
455 IF G=E(I) THEN 470
460 NEXT I
465 GOTO 485
470 LET H(I)=J
475 NEXT I2
480 GOTO 500
485 PRINT
490 PRINT "INVALID EMPLOYEE NUMBER - TRY AGAIN."
495 GOTO 440
500 PRINT
505 PRINT
```

```
510 PRINT
515 PRINT
520 PRINT "THE FOLLOWING IS A LIST OF REPORTS AVAILABLE"
525 PRINT "TO CHOOSE ONE TYPE IN THE # NEXT TO THE DESIRED REPORT"
530 PRINT
535 PRINT "     1 = PRINT PAYCHECKS"
540 PRINT "     2 = TABULATE PAYROLL RECORD"
545 PRINT "     3 = LIST EMPLOYEE RECORDS"
550 PRINT "     4 = SUMMERIZE EMPLOYER TAX RECORD"
555 PRINT "     5 = NONE"
560 PRINT
565 PRINT "WHICH ONE DO YOU WISH TO DO:";
570 INPUT R
575 PRINT
580 IF R=5 THEN 1030
585 IF R=1 THEN 610
590 IF R=2 THEN 975
595 IF R=3 THEN 1115
600 IF R=4 THEN 1270
605 GOTO 530
610 REM THIS SUB. DOES THE PAYCHECKS
615 PRINT
620 PRINT "WHEN THE CHECKS ARE IN PLACE TYPE A 7";
625 INPUT A
630 LET T1=5
635 LET T2=10
640 LET T3=15
645 LET T4=25
650 LET T5=5
655 LET T6=10
660 LET T7=15
665 LET T8=25
670 LET S1=50
675 LET S2=5
680 LET S3=60
685 REM THESE ARE USED TO SET THE PRINT TABS
690 REM THE PRINT FORMAT IS AS FOLLOWS:
695 REM        THIS IS THE CHECK PART
700 REM                                        DATE
705 REM      NAME                                    NET PAY
710 REM
715 REM       THIS IS FOR THE STUB PART
720 REM     EMP.#              EMP. NAME                HOURS WORKED
725 REM     GROSS PAY         FICA          FED. TAX          STATE TAX
730 REM     DATE              INS/DED        MISC.             NET PAY
735 REM
740 LET A=5
745 FOR I=1 TO A
750 PRINT
755 NEXT I
760 REM THIS SETS THE NUMBER OF LINES INITALLY SPACED FOR FIRST CHECK.
```

```
765 P5=0
770 P6=0
775 P7=0
780 P8=0
785 FOR I=1 TO H
790 PRINT TAB(S1);D$
795 PRINT
800 PRINT
805 REM THIS IS FOR THE SPACING BETWEEN THE DATE AND NAME LINE ON CHECK
810 PRINT TAB(S2);E$(I);
815 GOSUB 1475
820 GOSUB 1610
825 P4=R(I)*H(I)-P1-P2-P3-F(I)-C(I)
830 P4=INT(100*P4)/100
835 PRINT TAB(S3);P4
840 LET A=6
845 FOR I5=1 TO A
850 PRINT
855 NEXT I5
860 REM THE ABOVE IS FOR SPACING BETWEEN CHECK AND FIRST LINE ON STUB.
865 PRINT TAB(S2);E(I),E$(I),TAB(S3);H(I)
870 PRINT
875 PRINT
880 U1=R(I)*H(I)
885 U1=INT(100*U1)/100
890 PRINT TAB(T1);U1,TAB(T2);P1,TAB(T3);P2,TAB(T4);P3
895 PRINT
900 PRINT
905 PRINT TAB(T5);D$,TAB(T6);F(I),TAB(T7);C(I),TAB(T8);P4
910 LET A=8
915 FOR I5=1 TO A
920 PRINT
925 NEXT I5
930 REM A IS USED HERE TO SPACE TO THE NEXT CHECK DATE LINE.
935 P5=P5+P4
940 P6=P6+P2
945 P7=P7+P3
950 P8=P8+P1
955 NEXT I
960 P8=P8*2
965 REM P8 IS THE AMOUNT OF FICA THE EMPLOYER PAYS
970 GOTO 500
975 REM THIS SUB DOES THE PAYROLL RECORD
980 LET A=8
985 FOR I5=1 TO A
990 PRINT
995 NEXT I5
1000 PRINT "                    P A Y R O L L    R E C O R D   ";D$
1005 PRINT
1010 PRINT
1015 PRINT TAB(2);
```

```
1020 PRINT "        NAME   ","EMP# "," NET PAY "," TOT TAX"," TOT DED"
1025 PRINT
1030 FOR I=1 TO N
1035 GOSUB 1475
1040 GOSUB 1610
1045 PRINT E$(I),E(I),
1050 P4=H(I)*R(I)-P1-P2-P3-F(I)-C(I)
1055 Q1=P1+P2+P3
1060 Q2=F(I)+C(I)
1065 P4=INT(100*P4)/100
1070 PRINT P4,Q1,Q2
1075 Q3=Q3+P4
1080 Q4=Q4+Q1
1085 Q5=Q5+Q2
1090 NEXT I
1095 PRINT
1100 PRINT " TOTALS   ","                 ","",Q3,Q4,Q5
1105 PRINT
1110 GOTO 500
1115 REM THIS SUB GENERATES THE EMPLOYEE RECORDS STORED IN THIS PROGRAM
1120 P5=0
1125 P6=0
1130 P7=0
1135 P8=0
1140 LET A=10
1145 FOR I5=1 TO A
1150 PRINT
1155 NEXT I5
1160 PRINT "           E M P L O Y E E   PAY   R E C O R D S  ";Q6
1165 PRINT
1170 PRINT
1175 PRINT TAB(2);
1180 PRINT "  NAME   "," RATE "," DEPEND "," DED.  ","  MISC. "
1185 PRINT
1190 PRINT TAB(2);
1195 PRINT "  GR PAY "," FICA "," FED TAX"," ST TAX "," HR WK."
1200 PRINT
1205 PRINT
1210 FOR I=1 TO N
1215 U1=R(I)*H(I)
1220 U1=INT(100*U1)/100
1225 GOSUB 1475
1230 GOSUB 1610
1235 PRINT E$(I),R(I),D(I),F(I),C(I)
1240 PRINT U1,P1,P2,P3,H(I)
1245 PRINT
1250 NEXT I
1255 PRINT
1260 PRINT
1265 GOTO 500
1270 REM THIS SUB GENERATES THE EMPLOYER TAX RECORD
```

```
1275 LET A=10
1280 FOR I5=1 TO A
1285 PRINT
1290 NEXT I5
1295 PRINT "          EMPLOYER   TAX   RECORD   ";D$
1300 PRINT
1305 PRINT
1310 P5=0
1315 P6=0
1320 P7=0
1325 P8=0
1330 U1=0
1335 PRINT "TOT FICA "," TOT FED "," TOT ST "," TOT INS "," TOT MISC "
1340 PRINT TAB(6);
1345 PRINT "          "," TAX "," TAX "," DED "
1350 PRINT
1355 PRINT TAB(6);
1360 PRINT " GR PAY "," ST UNEM "," FED UNE"," MISC.   "
1365 PRINT
1370 PRINT
1375 FOR I=1 TO N
1380 GOSUB 1475
1385 GOSUB 1610
1390 P8=P8+P1
1395 P6=P6+P2
1400 P7=P7+P3
1405 P5=P5+F(I)
1410 P9=P9+C(I)
1415 U1=U1+R(I)*H(I)
1420 NEXT I
1425 P8=P8*2
1430 GOSUB 1695
1435 LET U1=INT(100*U1)/100
1440 LET U2=INT(100*U2)/100
1445 LET U3=INT(100*U3)/100
1450 PRINT P8,P6,P7,P5,P9
1455 LET U1=INT(100*U1)/100
1460 PRINT U1,U1,U2,U3
1465 PRINT
1470 GOTO 500
1475 REM THIS SUB CALCULATES THE FEDERAL TAX P2 AND FICA P1.
1480 REM THE INPUTS TO THIS SUB ARE R(I),D(I),H(I),H
1485 Z9=14100
1490 Z8=.0585
1495 REM THIS IS THE MAX FOR FICA WITHOLDING
1500 P1=R(I)*H(I)*Z8
1505 P1=INT(100*P1)/100
1510 Z7=R(I)*H(I)
1515 REM ********------------------------------
1520 IF Z7<Z9 THEN 1530
1525 P1=0
```

```
1530 REM THIS COMPLETES THE FICA CALC.
1535 REM NOW FOR THE FEDERAL TAX CALC.
1540 Z6=14.40
1545 Z6=Z6*H/40
1550 REM Z6 IS THE DED. ALLOWANCE
1555 Z6=Z6*D(I)
1560 Z5=Z7-Z6
1565 REM
1570 REM
1575 Z4=.1464
1580 REM *********--------------------------------------
1585 REM Z4 IS THE TAX RATE DUE
1590 P2=Z5*Z4
1595 IF P2>0 THEN 1600
1600 P2=INT(100*P2)/100
1605 RETURN
1610 REM THIS SUB GENERATES THE STATE WITHHOLDING TAX P3
1615 REM THE INPUT TO THIS SUB IS R(I),H(I),D(I)
1620 Z9=.15375
1625 Z8=33.30
1630 Z7=6.75
1635 REM Z8 & Z9 ARE THE FACTORS FOR THE STATE TAX
1640 Z1=R(I)*H(I)
1645 Z2=(12.5/40)*H
1650 Z2=Z2*D(I)
1655 Z3=Z8*H/40
1660 Z4=Z1-Z2-Z7
1665 P3=Z9*Z4-Z7
1670 IF P3>0 THEN 1685
1675 P3=0
1680 REM THIS FORMULA IS USED TO CALCULATE THE STATE WITHHOLDING TAX
1685 P3=INT(100*P3)/100
1690 RETURN
1695 REM THIS SUB CALCULATES THE EMPLOYER UNEMPLOYMENT TAX
1700 REM INPUT IS U1,H
1705 REM U1 IS THE STATE UNEM. & U2 IS THE FEDERAL UNEM. PORTION
1710 REM AND U3 IS A MISC. AMT. TO OFFER ROOM FOR FUTURE EXPANSION.
1715 LET U3=0
1720 REM THIS IS THE MISC. AMOUNT U3
1725 LET U2=4700
1730 REM THIS IS THE UNEMPLOYMENT CEILING PAYROLL
1735 U3=U2*H
1740 IF U1>U3 THEN 1785
1745 REM THIS IS FOR ACCUMULATED TOTALS
1750 U5=.03
1755 REM THIS IS THE STATE UNEM RATE INITIALLY
1760 U6=.01
1765 REM THIS IS THE FED ADDITION
1770 U1=U1*U5
1775 U2=U1*U6
1780 GOTO 1825
```

```
1785 REM THIS SUB SET THE CUTOFF FOR UNEM PAYMENTS
1790 IF W8=21 THEN 1815
1795 U1=U3*U5
1800 U2=U3*U6
1805 LET W8=21
1810 GOTO 1825
1815 LET U1=0
1820 LET U2=0
1825 RETURN
1830 END
```
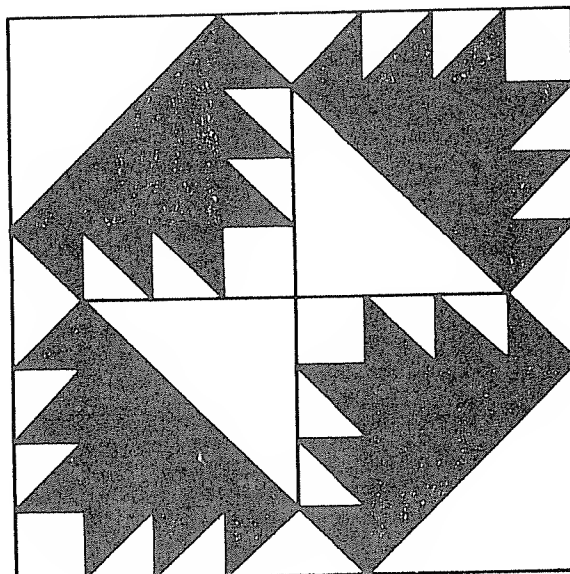
Payroll Modifications

There are only two modifications that may be made to enhance the operation of this program. They were not made before inclusion of the program in its present form because of the compatibility problems that exist between Advanced Basic compilers. The first and foremost modification would be to change the PRINT statements to PRINT USING statements in the report print outs. The second modification would be to use a Data File to feed information to the program instead of using internal DATA statements. If these modifications are not made the payroll program will still function normally and without error. These changes are primarily intended as a convenience factor, Not as a necessity.

The PRINT statements in all four of the reports may be converted to PRINT USING statements for added report clarity. The PRINT statements in the following lines should be converted: 835, 865, 890, 905, 1050, 1070, 1100, 1235, 1240, 1450 and 1460. The TAB( ) spacing for printing on the check and stub are listed on lines 630 to 680. By changing these tab constants the spacing between fields on the check and stub may be altered to conform to the particular check form in use.

The program may also be modified to allow the entry of employee data from a Use File rather then from the internal Data statements. This would require the addition of a FILES statements and conversion of the READ statements to READ # statements. The exact file modifications will depend on the computer system and Basic compiler version being used. If the program is modified for a Use File, delete the DATA statements in the program. For additional memory efficiency the PRINT and PRINT USING statements should be merged with the READ # statements. This merger should not be made unless the program has been modified for a Use File.

RUN

TYPE IN THE NUMBER OF HOURS IN THIS PAY PERIOD: 80

ENTER THE DATE (EX: 11-23-75): 12/24/75

DID ALL EMPLOYEES WORK THE FULL # OF HOURS (YES OR NO): NO

ENTER THE NUMBER OF EMPLOYEES THAT DIDN'T WORK THE FULL
NUMBER OF HOURS FOR THIS PAY PERIOD: 1

TYPE IN THE EMPLOYEE #, THEN # OF HOURS WORKED: AS ASKED
INPUT-EMP#, # HOURS WK: 3722,72

INVALID EMPLOYEE NUMBER - TRY AGAIN.
INPUT-EMP#, # HOURS WK: 3721,72

THE FOLLOWING IS A LIST OF REPORTS AVAILABLE
TO CHOOSE ONE TYPE IN THE # NEXT TO THE DESIRED REPORT

        1 = PRINT PAYCHECKS
        2 = TABULATE PAYROLL RECORD
        3 = LIST EMPLOYEE RECORDS
        4 = SUMMERIZE EMPLOYER TAX RECORD
        5 = NONE

WHICH ONE DO YOU WISH TO DO: 1

WHEN THE CHECKS ARE IN PLACE TYPE A ???

2/24/75

JOHN M. DOYLE   317.95

3721   JOHN M. DOYLE   72

470.15   27.5   68.39   56.81

2/24/75   7.5   0   317.95

2/24/75

G. R. CRAMER   189.15

6219   G. R. CRAMER   80

254.3   14.87   24.58   19.78

2/24/75   5.92   0   189.15

```
THE FOLLOWING IS A LIST OF REPORTS AVAILABLE
TO CHOOSE ONE TYPE IN THE # NEXT TO THE DESIRED REPORT

    1 = PRINT PAYCHECKS
    2 = TABULATE PAYROLL RECORD
    3 = LIST EMPLOYEE RECORDS
    4 = SUMMERIZE EMPLOYER TAX RECORD
    5 = NONE

WHICH ONE DO YOU WISH TO DO: ?2
```

```
             P A Y R O L L    R E C O R D    2/24/75


        NAME    EMP#             NET PAY        TOT TAX        TOT DED

   JOHN M. DOYLE        3721        317.95         144.7           7.5
   G. R. CRAMER         6219        189.15          59.23          5.92
   PHIL JOHNSON         1972        234.43         109.81          6.15

    TOTALS                          741.53         313.74         19.57
```
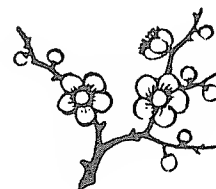
THE FOLLOWING IS A LIST OF REPORTS AVAILABLE
TO CHOOSE ONE TYPE IN THE # NEXT TO THE DESIRED REPORT

```
     1 = PRINT PAYCHECKS
     2 = TABULATE PAYROLL RECORD
     3 = LIST EMPLOYEE RECORDS
     4 = SUMMERIZE EMPLOYER TAX RECORD
     5 = NONE
```

WHICH ONE DO YOU WISH TO DO:?3

E M P L O Y E E   PAY   R E C O R D S   2-24-75

NAME	RATE	DEPEND	DED.	NET.
GR PAY	FICA	FED TAX	ST TAX	HP WH
JOHN M. DOYLE	6.50	2	7.5	
470.15	27.5	48.39	56.81	
G. R. CRAMER	3.1788	3	5.92	
254.3	14.87	24.58	19.79	
PHIL JOHNSON	4.38	1	6.15	
350.39	20.49	47.04	42.24	

THE FOLLOWING IS A LIST OF REPORTS AVAILABLE
TO CHOOSE ONE TYPE IN THE # NEXT TO THE DESIRED REPORT

    1 = PRINT PAYCHECKS
    2 = TABULATE PAYROLL RECORD
    3 = LIST EMPLOYEE RECORDS
    4 = SUMMERIZE EMPLOYER TAX RECORD
    5 = NONE

WHICH ONE DO YOU WISH TO DO:?4

E M P L O Y E R   T A X   R E C O R D   2/24/75

TOT FICA	TOT FED	TOT ST TAX	TOT INS TAX	TOT MISC DED
	GR PAY ST UNEM	FED UNE	MISC.	
125.72	132.05	118.09	15.57	
1074.86	37.24	18.74	0	

THE FOLLOWING IS A LIST OF REPORTS AVAILABLE
TO CHOOSE ONE TYPE IN THE # NEXT TO THE DESIRED REPORT

    1 = PRINT PAYCHECKS
    2 = TABULATE PAYROLL RECORD
    3 = LIST EMPLOYEE RECORDS
    4 = SUMMERIZE EMPLOYER TAX RECORD
    5 = NONE

WHICH ONE DO YOU WISH TO DO:?5

RISK:

This is a risk analysis program. It is used to analyze the probability of financial success regarding purchasing of stocks or new product sales. It considers the factors that will effect the cash flow for the investment.

USERS

Persons or companies who consider capital investments will find this program of considerable aid in determining the feasability of success.

INSTRUCTIONS

Most of the data will have to be entered into the program before execution. The program is self prompting and will ask for additional data as needed during execution. The program should be listed for additional information. The particular model of your system will have to be entered between lines 1000 and 2000.

LIMITATIONS

Line 200 contains DIM X(20,8), line 502 contains DEF FNR, and line 980 contains MAT = ZER statements. This program requires 10K Bytes of memory for storage. Thirty three kilo Bytes will be required for execution with the DIM statements set as shown in the listing. The program as shown contains all data and a model necessary to execute an example. This data and model will have to be removed before entering your data and/or investment model.

RISK


```
15REM
20REM
25REM     DESCRIPTION--RISK ANALYSIS PROGRAM BASED ON HERTZ
30REM                    SIMULATION MODEL DESCRIBED IN JAN-FEB
35REM                    1964   HBR ARTICLE.
40REM
45REM     INSTRUCTIONS-LINES 100-199 ARE RESERVED FOR DATA.
50REM                    LINES 985-2000 ARE RESERVED FOR THE MODEL
53REM
55REM                    ** A SAMPLE PROBLEM RUN EXISTS IN THE
60REM                    PROGRAM--TYPE RUN TO USE.....
65REM
70REM******************************************************************
80REM
100 DATA 100
101 DATA 15
102 DATA 0
103 DATA 9
104 DATA LIFE,5
105 DATA 5,8,10,12,15
106 DATA 0,.10,.40,.40,.10
107 DATA MARKET SIZE,5
108 DATA 100000,170000,250000,295000,340000
109 DATA 0,.10,.30,.40,.20
110 DATA MARKET GROWTH RATE,7
111 DATA 0,.01,.02,.03,.04,.05,.06
112 DATA 0,.05,.15,.30,.30,.15,.05
113 DATA SHARE OF MARKET,6
114 DATA .03,.06,.09,.12,.15,.17
115 DATA 0,.10,.15,.40,.30,.05
116 DATA SELLING PRICE,5
117 DATA 385,450,510,540,575
118 DATA 0,.10,.45,.35,.10
119 DATA OPERATING COSTS,6
120 DATA 370,400,435,470,505,545
121 DATA 0,.15,.35,.35,.10,.05
122 DATA INVESTMENT,6
123 DATA 7000000,8000000,9000000,9500000,10000000,10500000
124 DATA 0,.05,.15,.35,.35,.10
125 DATA RESIDUAL VALUE,4
126 DATA 3000000,4000000,4500000,5000000
127 DATA 0,.20,.40,.40
128 DATA FIXED COST,6
129 DATA 250000,275000,300000,325000,350000,375000
```

```
130 DATA 0,.10,.30,.30,.20,.10
200 DIM U(500),R(500),X(20,8),Y(20,8),X$(20),F(100),G(100)
210 DIM H(20),Z(500),C(20),C$(20)
212REM
214 LET Z8=1
216 LET Z9=.10
218REM
220 PRINT "DO YOU WANT TO USE DATA; (YES OR NO )" ;
230 INPUT Q1$
240 PRINT
250 IF Q1$="YES" THEN 290
260 LET Q9=P0=Q0=0
270 GO SUB 3000
280 GO TO 970
285REM          NUMBER OF ITERATIONS
290  READ N2
291REM     NUMBER OF INTERVALS IN HISTOGRAMS
292  READ S9
299REM    NUMBER OF CONSTANTS IN MODEL
300  READ N9
310 IF N9=0 THEN 350
320 FOR I = 1 TO N9
325REM     NAME OF CONSTANTS, VALUE OF CONSTANT
330 READ C$(I),C(I)
340 NEXT I
345REM    NUMBER OF CATEGORIES
350 READ N1
360 FOR I = 1 TO N1
365REM      NAME OF CATEGORIES, NO OF STEPS IN CATEGORY
370  READ X$(I),N(I)
380  FOR J = 1 TO N(I)
385REM    UPPER LIMITS OF STEPS
390 READ X(I,J)
400  NEXT J
405 LET P2=0
410  FOR J = 1 TO N(I)
415REM    PRIORS ON INTERVALS BETWEEN LIMITS
420 READ Y(I,J)
425 LET P2=P2+Y(I,J)
430  NEXT J
432 IF ABS(P2-1)<.0001 THEN 440
434 PRINT "
          SUM OF PRIORS FOR ";X$(I);" DOES NOT EQUAL 1"
435 STOP
440 NEXT I
445 GO TO 970
450REM     *****
500REM  FUNCTION-RANDOMLY SELECT A VALUE TO BE USED FOR
501REM             A PARTICULAR VARIABLE.
502 DEF FNR(M)
503 LET S1 = 0
```

```
504 LET R1 = RND(X)
505 FOR K9 = 1 TO N(M)
506 LET S1 = S1+Y(M,K9)
507 IF R1>=S1 THEN 510
508 LET FNR = X(M,K9-1) + (RND(X)*(X(M,K9)-X(M,K9-1)))
509 GO TO 511
510 NEXT K9
511 FNEND
515REM    *******
970 FOR I0 = 1 TO N2
980  MAT F = ZER
985REM ** MODEL IS ENTERED IN LINES 1000-1999  **
990REM    NO. OF YEARS OF LIFE
1000 LET N3 = INT(FNR(1) + .5)
1005REM   MARKET SIZE IN PERIOD 0
1010 LET A1=FNR(2)
1015REM   INVESTMENT CASH FLOW IN PERIOD 0
1020 LET F(0)=-1*FNR(7)
1030 LET G1=1
1035REM   FOR EACH YEAR OF LIFE
1040 FOR I=1 TO N3
1045REM   MULTIPLICATIVE GROWTH RATE  PERIOD I
1050 LET G1 = G1*(1+FNR(3))
1055REM   MARKET SIZE  PERIOD I
1060 LET A2 = A1*G1
1065REM   QUANTITY  PERIOD I
1070 LET A3 = A2*FNR(4)
1075REM     VALUE OF SALES  PERIOD I
1080 LET A4 = A3*FNR(5)
1085REM    OPERATING COSTS PERIOD I
1090 LET C1 = FNR(6)
1095REM    IS VALUE>=QTY * OP.COST
1100 IF A4>=A3*C1 THEN 1130
1110  LET A3 = 0
1120  LET A4 = 0
1125REM     TOTAL COST  PERIOD I
1130 LET A5 = FNR(9)+A3*C1
1135REM    CASH FLOW  PERIOD I
1140 LET F(I) = A4-A5
1150 NEXT I
1155REM    ADD RESID. VALUE TO CASH FLOW  LAST PERIOD
1160 LET F(N3) = F(N3)+FNR(8)
2000REM   ***** END OF MODEL *****
2010  FOR I = 0 TO N3
2015REM    PREPARE FOR RATE OF RETURN SUBROUTINE
2020 LET G(I+1) = F(I)
2030  NEXT I
2032  IF I0>1 THEN 2040
2034  GOSUB 8000
2040  GOSUB 5500
2050 NEXT I0
```

```
2060 GOSUB 6000
2070 IF Q1$="YES" THEN 99999
2080 LET Q9 = 1
2090 GOSUB 3000
2100 GO TO 970
3000REM    ** INPUT ROUTINE **
3002 IF Q9=1 THEN 3240
3010 READ Q,Q,Q
3020 PRINT "HOW MANY CONSTANTS IN YOUR MODEL";
3022 INPUT N9
3023 IF Q9=0 THEN 3040
3024 PRINT
3026 PRINT "ENTER THE VALUES FOR THE FOLLOWING:"
3028 PRINT
3030 FOR I = 1 TO N9
3032  READ C$(I),Q
3034  PRINT C$(I);
3036  INPUT C(I)
3038 NEXT I
3040 PRINT
3041 PRINT "HOW MANY RANDOM VARIABLE CATEGORIES IN YOUR MODEL";
3042 INPUT N1
3043 PRINT
3044 PRINT "      FOR THE FOLLOWING ITEMS THE USER SHOULD"
3045 PRINT "ENTER THE EXPECTED VALUE, THE LOWEST REASONABLE"
3046 PRINT "VALUE AND THE HIGHEST REASONABLE VALUE."
3050 PRINT "SEPARATE BY COMMAS."
3060 PRINT
3070 PRINT "      FOLLOWING THE USER'S ENTRIES A RANGE OF VALUES"
3080 PRINT "WILL BE TYPED OUT.  UNDER THIS WILL BE THE PERCENTAGE"
3090 PRINT "CHANCE THAT THE VALUE WILL FALL WITHIN A GIVEN RANGE."
3100 PRINT
3110 PRINT
3120 PRINT "***NOTE***IF THE USER IS UNHAPPY WITH THESE FIGURES,"
3130 PRINT "          HE WILL HAVE AN OPPORTUNITY TO CHANGE THEM"
3140 PRINT "          LATER ON."
3150 PRINT
3155 PRINT
3156 PRINT TAB(25);"EXPECTED, LOW, HIGH"
3157 PRINT
3158 LET Q2$="ABC"
3160 FOR I = 1 TO N1
3170  LET N(I) = 5
3175  LET Y(I,1) = 0
3180  READ X$(I),Q1
3190  FOR J = 1 TO Q1+Q1
3200   READ Q
3210  NEXT J
3220  GO SUB 4500
3230 NEXT I
3235 LET Q0=P0=0
```

```
3240 PRINT "DO YOU WANT TO CHANGE ANYTHING";
3250 INPUT Q2$
3260 PRINT
3270 IF Q2$="NO" THEN 3460
3275 LET Q0 = 1
3280 PRINT "INDEX NUMBER OF ITEM TO CHANGE";
3290 INPUT I
3300 PRINT
3310 PRINT "DO YOU WANT NEW VALUES";
3320 INPUT Q3$
3330 PRINT
3340 IF Q3$="NO" THEN 3380
3350 PRINT "INPUT 5 NEW VALUES";
3360 INPUT X(I,1),X(I,2),X(I,3),X(I,4),X(I,5)
3370 PRINT
3380 PRINT "DO YOU WANT NEW PERCENTAGES";
3390 INPUT Q3$
3400 PRINT
3410 IF Q3$="NO" THEN 3440
3415 LET P0 = 1
3420 PRINT "INPUT 4 NEW PERCENTAGES";
3430 INPUT Y(I,2),Y(I,3),Y(I,4),Y(I,5)
3432 LET P2=Y(I,2)+Y(I,3)+Y(I,4)+Y(I,5)
3433 IF ABS(P2-1)<.0001 THEN 3438
3434 PRINT "
               SUM OF PRIORS DOES NOT EQUAL 1"
3435 GOTO 3420
3438 PRINT
3440 GO SUB 4500
3450 GO TO 3240
3460 PRINT"HOW MANY TIMES DO YOU WANT TO ITERATE.   (0 WILL STOP PROGRAM)"
3470 INPUT N2
3490 PRINT
3500 IF N2=0 THEN 99999
3502 PRINT
3504 PRINT "HOW MANY INTERVALS DO YOU WANT IN THE HISTOGRAMS";
3506 INPUT S9
3508 PRINT
3510 RETURN
4500REM            ***MORE INPUT***
4505 IF Q0=1 THEN 4640
4520 PRINT
4530 PRINT I;X$(I);TAB(24);
4540 INPUT X(I,3),X(I,1),X(I,5)
4550 PRINT
4580 IF X(I,1)<1 THEN 4620
4590 LET X(I,2) = X(I,1)+INT(((X(I,3)-X(I,1))/2)+.5)
4600 LET X(I,4) = X(I,3)+INT(((X(I,5)-X(I,3))/2)+.5)
4610 GO TO 4640
4620 LET X(I,2) = X(I,1)+((X(I,3)-X(I,1))/2)
4630 LET X(I,4) = X(I,3)+((X(I,5)-X(I,3))/2)
```

```
4635REM       VALUES    ************************
4640 PRINT TAB(2);X(I,1);TAB(17);X(I,2);TAB(32);X(I,3);TAB(47);
4650 PRINT X(I,4);TAB(62);X(I,5)
4652 IF Q2$="YES" THEN 4780
4655 IF Q0=1 THEN 4690
4656 IF P0=1 THEN 4780
4660 LET Y(I,2)=Y(I,5)=.10
4690 LET Q2 = Y(I,2)*(X(I,1)+((X(I,2)-X(I,1))/2))
4700 LET Q3 = Y(I,5)*(X(I,4)+((X(I,5)-X(I,4))/2))
4710 LET Q4 = X(I,2)+((X(I,3)-X(I,2))/2)
4720 LET Q5 = X(I,3)+((X(I,4)-X(I,3))/2)
4730 LET Q6 = 1-Y(I,2)-Y(I,5)
4740 LET Q7 = Q4-Q5
4750 LET Q8 = (X(I,3)-Q2-(Q5*Q6)-Q3)/Q7
4760 LET Y(I,3) = Q8
4770 LET Y(I,4) = 1-Q8-Y(I,2)-Y(I,5)
4775REM       PERCENTS   ************************
4780 PRINT TAB(9);Y(I,2);TAB(24);Y(I,3);TAB(39);Y(I,4);TAB(54);
4790 PRINT Y(I,5)
4800 PRINT
4810 PRINT
4820 RETURN
5500REM          RATE OF RETURN AND PRESENT VALUE SUBROUTINES
5505REM          RATE OF RETURN
5506 IF Z8=0 THEN 5680
5510 LET R1=R2=R3=0
5530 LET R4 = INT(N3/2)
5540 LET R5 = R1
5550 FOR I = 1 TO N3+1
5560   LET R6 = I-R4
5570   LET R7 = -1*R6
5580   LET R2 = (G(I)*(EXP(1)↑(R5*R7)))+R2
5590   LET R3 = ((R7*G(I))*(EXP(1)↑(R5*R7)))+R3
5600 NEXT I
5610 LET R1 = R5-(R2/R3)
5620 LET R2=R3=0
5630 LET R8 = R1-R5
5640 IF ABS(R8)>.000005 THEN 5540
5650 LET R(I0) = (EXP(1)↑R1)-1
5670 LET R(I0) = INT(10000*R(I0)+.5)/10000
5680REM          NET PRESENT VALUE
5690 LET R1=Z1=Z2=0
5700 FOR I = 0 TO N3
5710   LET R1 = R1+(F(I)/((1+Z9)↑I))
5711   IF F(I)>0 THEN 5714
5712   LET Z1 = Z1+(ABS(F(I))/((1+Z9)↑I))
5713   GO TO 5720
5714   LET Z2 = Z2+(F(I)/((1+Z9)↑I))
5720 NEXT I
5730 LET V(I0) = R1
5735 LET Z(I0) = INT(10000*(Z2/Z1)+.5)/10000
```

```
5740 RETURN
5750REM    ********            ******
6000REM       SET UP FOR HISTOGRAM PRINT SUBROUTINE
6002 IF Z9=0 THEN 6050
6010 PRINT "RATES OF RETURN FOR";N2;"SAMPLES"
6020 PRINT
6030 LET I1 = 1
6040 GO SUB 7000
6050 PRINT "NET PRESENT VALUES FOR";N2;"SAMPLES"
6060 PRINT
6070 LET I1 = 2
6079 MAT R=ZER
6080 MAT R = R+V
6090 GO SUB 7000
6100 PRINT "RATIOS OF NET PRESENT VALUE FOR";N2;"SAMPLES"
6110 PRINT
6120 LET I1=3
6129 MAT R=ZER
6130 MAT R = R+Z
6140 GO SUB 7000
6150 RETURN
7000REM       PRINT HISTOGRAMS SUBROUTINE
7010 LET R1=R2=R(1)
7012 LET A9$ = "*"
7020 FOR I = 2 TO N2
7030   IF R(I)=R1 THEN 7110
7040   IF R(I)<R1 THEN 7080
7050   IF R(I)=R2 THEN 7110
7060   IF R(I)>R2 THEN 7100
7070   GO TO 7110
7080   LET R1 = R(I)
7090   GO TO 7110
7100   LET R2 = R(I)
7110 NEXT I
7120 ON I1 GO TO 7130,7150,7130
7130 LET R3 = INT(1000*((R2-R1)/(S9-1))+.5)/1000
7140 GO TO 7160
7150 LET R3 = INT(((R2-R1)/(S9-1))+.5)
7152 IF R2<99999 THEN 7160
7154 PRINT "FOR ACTUAL VALUE MULTIPLY BY 1000"
7156 PRINT
7160 MAT H = ZER
7170 LET S4=S5=H1=0
7180 LET S1 = R1
7190 FOR I = 1 TO S9
7200   LET S2 = S1+R3
7210   FOR J = 1 TO N2
7220     IF R(J)>=S2 THEN 7270
7230     IF R(J)<S1 THEN 7270
7240     LET H(I) = H(I)+1
7250     LET S4 = S4+R(J)
```

```
7260    LET S5 = S5+R(J)↑2
7270  NEXT J
7272    ON I1 GO TO 7274,7278,7274
7274    PRINT INT(S1*10↑4+.5)/10↑4;" LESS THAN";INT(S2*10↑4+.5)/10↑4;
7276    GO TO 7282
7278    IF R2<99999 THEN 7281
7279    PRINT INT(S1/1000+.5);" LESS THAN";INT(S2/1000+.5);
7280    GO TO 7282
7281    PRINT INT(S1+.5);" LESS THAN";INT(S2+.5);
7282    PRINT TAB(31);
7283    IF H(I)=0 THEN 7288
7284    FOR J = 1 TO H(I)
7285     PRINT A9$;
7286    NEXT J
7288    PRINT TAB(55);H(I)
7290    LET S1 = S2
7300    LET H1 = H1+H(I)
7310  NEXT I
7320  PRINT
7330  PRINT
7340  ON I1 GO TO 7350,7370,7350
7350  PRINT "MEAN";INT(10000*(S4/H1)+.5)/10000
7360  GO TO 7380
7370  PRINT "MEAN";S4/H1
7380  PRINT
7390  LET V4 = (S5/H1)-((S4/H1)↑2)
7400  PRINT "STANDARD DEVIATION";INT(10000*SQR(V4)+.5)/10000
7410  PRINT
7420  PRINT
7430  PRINT
7440  RETURN
8000 REM      PRINT FIRST SET OF FLOWS
8010  PRINT "FLOWS FOR THE FIRST ITERATION FOR";N3+1;" PERIODS"
8020  PRINT
8030  PRINT "PERIOD","FLOW"
8040  PRINT
8050  FOR I = 0 TO N3
8060    PRINT I,F(I)
8070  NEXT I
8080  PRINT
8090  PRINT
8100  RETURN
99999 END
```

ENTERING DATA

Lines 100 thru 199 are reserved for entering your data into the program. In
the program's source code listing lines 100 thru 130 contain sample data,
this data must be deleted from the program before entering your data. Most
of the data must be entered in DATA statements before the program is run,
however a built in routine permits some data to be input during execution
of the program.

The sample program cut shown here and the definitions immediately following
are provided as an aid to understanding the program and the data entry format.

```
*LIST 100-130

100 DATA 100
101 DATA 15
102 DATA 0
103 DATA 9
104 DATA LIFE,5
105 DATA 5,8,10,12,15
106 DATA 0,.10,.40,.40,.10
107 DATA MARKET SIZE,5
108 DATA 100000,170000,250000,295000,340000
109 DATA 0,.10,.30,.40,.20
110 DATA MARKET GROWTH RATE,7
111 DATA 0,.01,.02,.03,.04,.05,.06
112 DATA 0,.05,.15,.30,.30,.15,.05
113 DATA SHARE OF MARKET,6
114 DATA .03,.06,.09,.12,.15,.17
115 DATA 0,.10,.15,.40,.30,.05
116 DATA SELLING PRICE,5
117 DATA 385,450,510,540,575
118 DATA 0,.10,.45,.35,.10
119 DATA OPERATING COSTS,6
120 DATA 370,400,435,470,505,545
121 DATA 0,.15,.35,.35,.10,.05
122 DATA INVESTMENT,6
123 DATA 7000000,8000000,9000000,9500000,10000000,10500000
124 DATA 0,.05,.15,.35,.35,.10
125 DATA RESIDUAL VALUE,4
126 DATA 3500000,4000000,4500000,5000000
127 DATA 0,.20,.40,.40
128 DATA FIXED COST,6
129 DATA 250000,275000,300000,325000,350000,375000
130 DATA 0,.10,.30,.30,.20,.10
```

THE MODEL

Lines 1000 thru 1999 are reserved for entering your model. The sample cut shown here and the explanations following it have been provided to aid you in programming your own model. Substitute your model for the one contained in the program before running.

```
*LIST 985-2000

985REM ** MODEL IS ENTERED IN LINES 1000-1999  **
990REM    NO. OF YEARS OF LIFE
1000 LET N3 = INT(FNR(1) + .5)
1005REM   MARKET SIZE IN PERIOD 0
1010 LET A1 = FNR(2)
1015REM   INVESTMENT CASH FLOW IN PERIOD 0
1020 LET F(0) = -1*FNR(7)
1030 LET G1 = 1
1035REM   FOR EACH YEAR OF LIFE
1040 FOR I = 1 TO N3
1045REM    MULTIPLICATIVE GROWTH RATE  PERIOD I
1050 LET G1 = G1*(1+FNR(3))
1055REM   MARKET SIZE  PERIOD I
1060 LET A2 = A2*G1
1065REM   QUANTITY  PERIOD I
1070 LET A3 = A2*FNR(4)
1075REM   VALUE OF SALES  PERIOD I
1080 LET A4 = A3*FNR(5)
1085REM   OPERATING COSTS  PERIOD I
1090 LET C1 = FNR(6)
1095REM   IS VALUE>=QTY * OP. COST
1100 IF A4>=A3*C1 THEN 1130
1110 LET A3 = 0
1120 LET A4 = 0
1125REM   TOTAL COST  PERIOD I
1130 LET A5 = FNR(9)+A3*C1
1135REM   CASH FLOW  PERIOD I
1140 LET F(I) = A4-A5
1150 NEXT I
1155REM   ADD RESID. VALUE TO CASH FLOW  LAST PERIOD
1160 LET F(N3) = F(N3)+FNR(8)
2000REM    ****** END OF MODEL  *****
```

605

PROGRAM DATA DEFINED:

Line 100:   # of iterations or trials to be used to generate the cash flows.

Line 101:   # of discrete increments wanted in histogram. 100 iterations would require somewhere between 12 and 20; the example uses 15.

Line 102:   # of constants or nonrandom variables in the model. This number represents the total number of items in the model whose values remain unchanged in each set of cash flows generated.

Line 103:   # of random variable categories in the model. This number represents the total number of items in the model whose ultimate values will be randomly selected by the program.

Line 104:   Name of the first variable factor, # of steps for this factor (never more than 8). The example uses 5.

Line 105:   The upper limits for each step. There must be an equivalent number of values entered as there are steps in the preceding line - in this case there must be 5.

Line 106:   Estimated chance expressed in percentages of the future value falling into each step. The first number must always be a $\emptyset$. The chances when summed should equal 100%. Counting the $\emptyset$ as one, the total number of chances must be equal to the number of values entered in the preceding line - for the example this is 5. The data line in the example reads: a $\emptyset$% chance of life being less then 5, a 10% chance of its being between 5 and 8, a 40% chance of its being between 8 and 10, a 40% chance of its being between 10 and 12, and a 10% chance of its being between 12 and 15.

Line 107:   See line 104 above. Lines 104 thru 106 should be repeated for as many random variable factors as called for in line 103. In this example this is 9.

NOTE: If the data item in line 102 is a number other than $\emptyset$, the user will have data statements on those constants starting in line 103 and continuing for a corresponding number of lines thereafter. The data that now appears starting in line 103 would then begin in the first line after the data statements for the constants. Suppose for example this particular problem contained 2 constants, months equal to 12 and purchase price equal to 250. Lines 102 thru 104 would then read:

                    102 DATA, 2
                    103 DATA MONTHS, 12
                    104 DATA PURCHASE PRICE, 250

The data now in line 103 and the data in the subsequent lines would now begin with line 105.

REMEMBER: Data must be entered in DATA statements, all DATA statements are read and the names of the constants and the random variable categories are used in a questioning process.

MODEL PROGRAMMING DATA

Lines 200 and 210 contain the dimensioned variables used in this model. When programming your model these variables, with the exception of F(I) should be deleted.

Variables that have not been used in the program and are available for the user are: A∅, A1, A2, ..., A9; B∅, B1, B2, ..., B9; and C∅, C1, C2, ..., C9.

The subscripted "F" represents the cash flow for a given or calculated period. Your program must be written so as to end up with a value for F(∅) - cash flow in period ∅ and values for F(1) through F(N3) - cash flow in period 1 through cash flow in last period.

Your model must calculate a value for "N3", the number of periods of cash flow, excluding period ∅.

FNR(M) is a function which selects a value for a random variable factor given your value intervals and probabilities for that factor. In place of the "M" in FNR(M) substitute the number or index of the factor whose value you want to calculate. In the example there are 9 random variable factors contained in the DATA statements; Life is #1, Market Size is #2, Market Growth Rate is #3, Share of Market is #4, etc. Thus the statement in the program model, line 1010 which reads LET A1 = FNR(2) sets A1 equal to some Market Size, which falls within one of the intervals set by you. This function is completely defined in lines 500 thru 511, where X(MK,9) is the actual value for factor M; step K9 - and Y(M,K9) is the probability for category M; step K9, K9 = 1 to the number of steps in factor "M".

Model Location - When the model is confined to lines 985 thru 1999, it is automatically embedded in the loop - 970 FOR I∅ = 1 TO N2 - where, N2 is the number of iterations specified in DATA line 100; in the example this is 1∅∅. It is necessary to program the model as if it were to be run only once, otherwise the iteration procedure is effected. Use the variables I, J, K and L with the FOR-NEXT statements in the model.

To demonstrate more clearly how to go about programming your own model an explanation of what the model in this example is calculating is included here. A new product line is to be analyzed and management has provided the data for the 9 key factors shown in DATA lines 104 thru 130. The relationships among these variables is described below:

Line 1000: The number of periods is computed by rounding off, using the intergerizing function INT. Life is the first category therefore, FNR(1) provides the random draw for Life.

Line 1010: The Market Size in period 0 - A1, is set equal to the random draw for Market Size; the second category.

Line 1020: The cash flow in period 0 - F(0), is set equal to the negative of the random draw for Investment; the seventh category. Note: If the values in line 123 had been entered as negatives then line 1020 would read - 1020 LET F(0) = FNR(7).

Line 1030: This line initializes the variable G1 to 1, which represents the Market Growth Rate in each period and is considered to be 1 in period 0.

Line 1040: The cash flows F(I) are computed for each period, from 1 to N3, in a loop which extends from line 1040 to line 1150. The "I" represents the index number of the period.

Line 1050: The multiplicative Market Growth Rate in period 1 - G1, is set equal to the growth rate in the previous period times one plus the Market Growth Rate in period I; the third category.

Line 1060: The total Market Size in period I - A2, is set equal to the Market Size in period 0 - A1, times the multiplicative Market Growth Rate.

Line 1070: The quantity to be sold by the company in period I - A3, is set equal to the total Market Size - A2, times the random draw for Share of Market; the fourth category.

Line 1080: The total value of the sales of the company in period I - A4, is set equal to the quantity sold - A3, times the random draw for Selling Price; the fifth category.

Line 1090: The operating cost per item in period I - C1, is set equal to the random draw for Operating Costs; the sixth category.

Lines 1100 - 1120: If the total value of the sales by the company - A4, are greater than or equal to the total operating costs - A3 times C1, then the company will produce the product in period I and both the quantity to be sold - A3, and the total value of sales - A4, will retain their values previously computed. If the total value of the sales - A4, is less than the total operating costs - A3 times C1, then both A4 and A3 are set equal to 0, i.e., in period I the company will not produce or sell any of the product thereby avoiding all marginal costs.

Line 1130:   The total costs in period I - A5, is set equal to the random draw for Fixed Cost; the ninth category, plus the total operating costs - A3 times C1.

Line 1140:   The cash flow in period I - F(I), is set equal to the value of sales in period I - A4, minus the total period costs - A5.

Line 1160:   The cash flow in the last period - F(N3), is set equal to the value previously computed in line 1140 for F(N3), plus the random draw for Residual Value; the eighth category.

Modifications:

It may be necessary to omit or remove the rate of return routine because it is often programmed to simply to allow for all the solutions that are possible when calculating the rate of return. If your model generates a set of cash flows that never change sign - the error message OVERFLOW will appear indicating the need to either reprogram the model or eliminate the rate of return routine. This routine may be eliminated by changing line 214 to read: LET Z8=0.

In the example the net present values are set to be discounted at a rate of 10%. If a different rate is desired, change line 216 from: LET Z9=.10 to the discount rate desired. If discounting is not wanted, change line 216 to read: LET Z9=0.

An example of the output of RISK is shown on the next two pages. The output starts with the set of cash flows generated during the first iteration (additional outputs have not been shown due to the amount of space that would be required to show the remaining iterations). A quick check of the first set of cash flows will allow you to determine if the figures seem reasonable. Histograms for the Rates of Return, Net Present Values and Ratios of Net Present Values follow the cash flow and following the histogram are the Mean and Standard Deviation for the particular measure plotted in the histogram.

```
*RUN

DO YOU WANT TO USE DATA? (YES OR NO)   ?YES

FLOWS FOR THE FIRST ITERATION FOR  12 PERIODS

    PERIOD                 FLOW

        0              -9216754
        1               3227312
        2               1029929
        3              -2882727.8
        4               1326603
        5               2133060
        6               3264069
        7               5674858
        8                478938.6
        9                981559.9
       10               4040488
       11               9438514

RATES OF RETURN FOR  100 SAMPLES

 -.011   LESS THAN   .022      ***                                  3
  .022   LESS THAN   .055      ******                               6
  .055   LESS THAN   .088      ***********                          11
  .088   LESS THAN   .121      ************                         12
  .121   LESS THAN   .154      ******************                   18
  .154   LESS THAN   .187      *******************                  19
  .187   LESS THAN   .22       ***********                          11
  .22    LESS THAN   .253      *******                              7
  .253   LESS THAN   .286      ******                               6
  .286   LESS THAN   .319      ****                                 4
  .319   LESS THAN   .352      **                                   2
  .352   LESS THAN   .385                                           0
  .385   LESS THAN   .418                                           0
  .418   LESS THAN   .451      *                                    1
  .451   LESS THAN   .484                                           0


MEAN      .1585

STANDARD DEVIATION      .0809
```

```
NET PRESENT VALUES FOR  100 SAMPLES

FOR ACTUAL VALUE, MULTIPLY BY 1000

   -5391    LESS THAN    -3962      ***                                      2
   -3962    LESS THAN    -2533      *******                                  6
   -2533    LESS THAN    -1104      ***********                             10
   -1104    LESS THAN      325      *******                                  6
     325    LESS THAN     1754      ****************                        16
    1754    LESS THAN     3183      ************                            12
    3183    LESS THAN     4612      ***********                             11
    4612    LESS THAN     6041      ****************                        16
    6041    LESS THAN     7470      ********                                 8
    7470    LESS THAN     8899      *****                                    5
    8899    LESS THAN    10328      ***                                      3
   10328    LESS THAN    11756      *                                        1
   11756    LESS THAN    13185      **                                       2
   13185    LESS THAN    14614      *                                        1
   14614    LESS THAN    16043      *                                        1


MEAN      3101307

STANDARD DEVIATION      4108050
```

```
RATIOS OF NET PRESENT VALUE FOR  100 SAMPLES

   .4592    LESS THAN    .6222      **                                       2
   .6222    LESS THAN    .7852      *********                                9
   .7852    LESS THAN    .9482      *********                                9
   .9482    LESS THAN   1.1112      ***********                             11
  1.1112    LESS THAN   1.2742      ******************                      18
  1.2742    LESS THAN   1.4372      ************                            12
  1.4372    LESS THAN   1.6002      ************                            12
  1.6002    LESS THAN   1.7632      ***********                             11
  1.7632    LESS THAN   1.9262      *********                                9
  1.9262    LESS THAN   2.0892      **                                       2
  2.0892    LESS THAN   2.2522      ***                                      3
  2.2522    LESS THAN   2.4152                                               0
  2.4152    LESS THAN   2.5792      *                                        1
  2.5792    LESS THAN   2.7412                                               0
  2.7412    LESS THAN   2.9042      *                                        1


MEAN     1.3327

STANDARD DEVIATION      .4334
```

SCHEDULE 2:

DESCRIPTION

Schedule 2 computes the best assignment of personnel or resources to locations. This program can also determine the most efficient scheduling of jobs in a machine shop or men to various jobs.

USERS

This program will be useful to individuals who have to schedule resources or the flow of work through an office.

INSTRUCTIONS

Your data must be entered into data statements before the program is run. The program is documented and should be listed for instructions. Sample data is contained in lines 6020 to 6190 and should be removed before entering your data.

LIMITATIONS

This program requires matrix manipulations. In line 150 is a DIM A(20,5) statement and line 470 contains a Print Using statement. It requires 34K Bytes of on line memory for storage and execution. The execution length may be reduced by lowering the size of the DIM statements in lines 130 to 170. The program source code requires 12K Bytes for storage.

```
110 PRINT"GEOMETRIC SIMULATOR"
120 PRINT
130 DIM B(15)
140 DIM R(15)
150 DIM A(20,5),D(15),T(15),F(15)
160 DIM S(15,15),P(15,15),M(16,16),G(15,15)
170 DIM U(15,15),W(15,15),W(15,15),Z(15,15)
180 READ N,M
190 IF N <> 0 THEN 390
200 READ N,M
210 PRINT"INSTRUCTIONS:"
220 PRINT"THIS PROGRAM CALULATES A JOB SHOP SCHEDULE USING"
230 PRINT"A HEURISTIC GEOMETRIC METHOD."
240 PRINT"ENTER DATA IN THE FOLLOWING ORDER BEGINNING"
250 PRINT"IN LINE 6000:"
260 PRINT"    * THE NUMBER OF JOBS TO BE PROCESSED,"
270 PRINT"    * THE NUMBER OF MACHINES,"
280 PRINT"    * THE SEQUENCE MATRIX S(I,J) BY ROWS"
290 PRINT"      WHERE S(I,J) IS THE SEQUENCE OF THE I'TH"
300 PRINT"      JOB ON THE J'TH MACHINE,"
310 PRINT"    * THE PROCESS TIME MATRIX P(I,J) BY ROWS"
320 PRINT"      WHERE P(I,J) IS THE PROCESS TIME REQUIRED"
330 PRINT"      BY THE I'TH JOB ON THE J'TH MACHINE."
340 PRINT
350 PRINT"SAMPLE DATA IS ALREADY IN LINES 6000-6190."
360 PRINT
370 PRINT"TO EXECUTE THE PROGRAM:"
380 PRINT"    TYPE 'DELETE 6000-6190' (DELETES SAMPLE DATA)"
390 PRINT"    ENTER YOUR DATA"
400 PRINT"    TYPE 'RUN'"
410 PRINT
420 PRINT"THIS PROGRAM REQUIRES LARGE AMOUNTS OF PROCESSOR TIME"
430 PRINT
440 PRINT"THE FOLLOWING IS A SAMPLE EXECUTION USING THE"
450 PRINT"SAMPLE DATA."
460 PRINT
470 PRINT USING 480,N,M
480:SCHEDULING ##-JOBS ON ##-MACHINES
490 FOR I=1TON
500 FOR J=1TOM
510 READ S(I,J)
520 LET M(I+1,1+J)=0.
530 NEXT J
```

```
540 NEXT I
550 FOR I=1TON
560 FOR J=1TOM
570 READ P(I,J)
580 NEXT J
590 NEXT I
600 PRINT"INPUT"
610 PRINT"SEQUENCE MATRIX-S"
620 PRINT
630 GOSUB 650
640 GO TO 720
650 PRINT"MACHINE   1 ";
660 FOR I=2 TO M
670 PRINT TAB(I*5);I;
680 NEXT I
690 PRINT
700 PRINT"JOB"
710 RETURN
720 FOR I=1TON
730 PRINT USING 740,I;
740:##
750 FORJ=1TOM
760 PRINT TAB(5.*J);S(I,J);
770 NEXT J
780 PRINT
790 NEXT I
800 PRINT
810 PRINT
820 PRINT"PROCESSING TIME MATRIX-P"
830 PRINT
840 GOSUB 650
850 FOR I=1TON
860 PRINT USING 740,I;
870 FOR J=1TOM
880 PRINT TAB(5.*J);P(I,J);
890 NEXT J
900 PRINT
910 NEXT I
920 GOTO 1070
930 PRINT
940 PRINT
950 PRINT"OUTPUT SCHEDULE"
960 PRINT
970 PRINT"C7=RULE"
980 PRINT"C=NO. OF ITERATIONS"
990 PRINT TAB(16);"INITIAL COND.";TAB(42);"ITERATIONS"
1000 PRINT TAB(12);"TMIN";TAB(25);"TAVE";
1010 PRINT TAB(38);"TMIN";TAB(51);"TAVE"
1020 PRINT "R.S.";TAB(12);"A(1,1)";TAB(25);"A(1,2)";
1030 PRINT TAB(38);"A(2,1)";TAB(51);"A(2,2)"
1040 PRINT "L.S.";TAB(12);"A(3,1)";TAB(25);"A(3,2)";
```

```
1050 PRINT TAB(38);"A(4,1)";TAB(51);"A(4,2)"
1060 REM CALC TOTAL PROCESS TIME FOR JOBS
1070 FOR I=1 TO N
1080 LET P(I,M+1.)=0.
1090 FOR J=1 TOM
1100 LET P(I,M+1.)=P(I,M+1.)+P(I,J)
1110 NEXT J
1120 NEXT I
1130 LET E=2.
1140 LET C1=0.
1150 LET C2=1.
1160 LET T=9000.
1170 LET C3=T
1180 LET C6=0.
1190 LET C7=1.
1200 LET L=1.
1210 LET X1=1.
1220 LET X2=0.
1230 LET X3=0.
1240 LET X4=0.
1250 LET A(2,3)=T
1260 REM GENERATE M(I+1,1+J)
1270 FOR I=0TON
1280 FOR J=0TOM
1290 LET M(I+1,1+S(I,J))=J
1300 NEXT J
1310 NEXT I
1320 REM GENERATE MAX SEQ NO. IN JOBS
1330 FOR I=1TON
1340 LET S=0.
1350 FOR J=1TOM
1360 IF S(I,J)<=S THEN 1380
1370 LET S=S(I,J)
1380 NEXT J
1390 LET B(I)=S
1400 NEXT I
1410 LET A(1,1)=T
1420 LET A(2,1)=T
1430 LET A(3,1)=T
1440 LET A(4,1)=T
1450 LET A(1,3)=1.
1460 LET A(1,2)=T
1470 LET A(2,2)=T
1480 LET A(3,2)=T
1490 LET A(4,2)=T
1500 LET C6=0.
1510 FOR H=1TON
1520 LET C5=0.
1530 LET C=1.
1540 IF C6=1. THEN 2320
1550 FOR I=1TON
```

```
1560 FOR J=1TON
1570 LET G(I,J)=0.
1580 NEXT J
1590 NEXT I
1600 LET B3=T
1610 REM RIGHT SHIFT
1620 FOR I=1TON
1630 LET D(I)=B(I)
1640 LET F(I)=B3
1650 LET T(I)=B3
1660 LET R(I)=P(I,M+1.)
1670 NEXT I
1680 IF C5=1. THEN 1750
1690 LET Y=M(H+1,1+D(H))
1700 LET R(H)=R(H)-P(H,Y)
1710 LET G(H,Y)=T-P(H,Y)
1720 LET F(H)=G(H,Y)
1730 LET T(H)=F(H)
1740 LET D(H)=D(H)-1.
1750 LET C4=1.
1760 IF L=2. THEN 5110
1770 FOR I=1TON
1780 LET Y=M(I+1,1+D(I))
1790 IF Y>0. THEN 1820
1800 LET T(I)=0.
1810 GO TO 1890
1820 FOR K=1TON
1830 IF K=I THEN 1880
1840 IF T(I)-P(I,Y) >= G(K,Y)+P(K,Y) THEN 1880
1850 IF T(I)<=G(K,Y) THEN 1880
1860 LET T(I)=G(K,Y)
1870 GO TO 1820
1880 NEXT K
1890 NEXT I
1900 REM FIND MAX PRIORITY FOR RULES WITH T(I)
1910 LET J=0.
1920 LET S=0.
1930 FOR I=1TON
1940 LET Y=M(I+1,1+D(I))
1950 LET Z1=X1*T(I)-X2*P(I,Y)+X3*P(I,Y)+X4*R(I)
1960 IF Z1<S THEN 2010
1970 IF Z1>S THEN 1990
1980 IF P(J,M(J+1,1+D(J)))<=P(I,M(I+1,1+D(I))) THEN 2010
1990 LET S=Z1
2000 LET J=I
2010 NEXT I
2020 IF S=0. THEN 2160
2030 REM UPDATE
2040 LET Y=M(J+1,1+D(J))
2050 IF L=3. THEN 2080
2060 LET G(J,Y)=T(J)-P(J,Y)
```

```
2070 GO TO 2090
2080 LET G(J,Y)=S
2090 LET F(J)=G(J,Y)
2100 LET R(J)=R(J)-P(J,Y)
2110 FOR I=1TON
2120 LET T(I)=F(I)
2130 NEXT I
2140 LET D(J)=D(J)-1.
2150 GO TO 1770
2160 REM RESET G(I,J) TO ZERO START TIME
2170 REM FIND MIN G(I,M(I+1,1+1))
2180 LET S=T
2190 FOR I=1TON
2200 IF G(I,M(I+1,1+1))>=S THEN 2220
2210 LET S=G(I,M(I+1,1+1))
2220 NEXT I
2230 FOR I=1TON
2240 FOR J=1TOM
2250 LET G(I,J)=G(I,J)-S
2260 NEXT J
2270 NEXT I
2280 REM END (450)
2290 REM END RIGHT SHIFT
2300 GO TO 2940
2310 REM LEFT SHIFT
2320 LET C4=2.
2330 FOR I=1TON
2340 FOR J=1TOM
2350 LETG(I,J)=T
2360 NEXT J
2370 NEXT I
2380 FOR I=1TON
2390 LET D(I)=1.
2400 LET F(I)=0.
2410 LET T(I)=0.
2420 LET R(I)=P(I,M+1.)
2430 NEXT I
2440 IF C5=1. THEN 2510
2450 LET Y=M(H+1,1+1)
2460 LET R(H)=R(H)-P(H,Y)
2470 LET G(H,Y)=0.
2480 LET F(H)=P(H,Y)
2490 LET D(H)=D(H)+1.
2500 LET T(H)=F(H)
2510 LET C4=2.
2520 IF L=2. THEN 5320
2530 FOR I=1 TO N
2540 IF D(I)<=B(I) THEN 2570
2550 LET T(I)=T
2560 GO TO 2650
2570 LET Y=M(I+1,1+D(I))
```

```
2580 FOR K=1TON
2590 IF K=I THEN 2640
2600 IF T(I)+P(I,Y)<=G(K,Y) THEN 2640
2610 IF T(I)>=G(K,Y)+P(K,Y) THEN 2640
2620 LET T(I)=G(K,Y)+P(K,Y)
2630 GO TO 2580
2640 NEXT K
2650 NEXT I
2660 REM FIN MIN PRIORITY FOR RULES WITH T(I)
2670 LET J=0.
2680 LET S=T
2690 FOR I=1 TON
2700 LET Y=M(I+1,1+D(I))
2710 LET Z1=X1*T(I)+X2*P(I,Y)+X3*(1000.-P(I,Y))+X4*(4000.-R(I))
2720 IF Z1>S THEN 2770
2730 IF Z1<S THEN 2750
2740 IF P(J,M(J+1,1+D(J))) <= P(I,M(I+1,1+D(I))) THEN 2770
2750 LET S=Z1
2760 LET J=I
2770 NEXT I
2780 IF S=T THEN 2940
2790 REM UPDATE
2800 LET Y=M(J+1,1+D(J))
2810 IF L=3. THEN 2840
2820 LET G(J,Y)=T(J)
2830 GO TO 2850
2840 LET G(J,Y)=S-P(J,Y)
2850 LET R(J)=R(J)-P(J,Y)
2860 LET F(J)=G(J,Y)+P(J,Y)
2870 FOR I=1TON
2880 LET T(I)=F(I)
2890 NEXT I
2900 LET D(J)=D(J)+1.
2910 GO TO 2520
2920 REM END LEFT SHIFT
2930 GO TO 2940
2940 REM CALC A(I,J)
2950 LET B1=0.
2960 LET B2=0.
2970 FOR I=1TON
2980 LET S=G(I,M(I+1,1+B(I)))+P(I,M(I+1,1+B(I)))
2990 IF S<=B1 THEN 3010
3000 LET B1=S
3010 LET B2=B2+S
3020 NEXT I
3030 LET B2=B2/N
3040 IF E=2. THEN 3060
3050 GO TO 3150
3060 IF B1>=A(2,3) THEN 3150
3070 LET A(2,3)=B1
3080 LET A(3,3)=B2
```

```
3090 LET A(4,3)=C7
3100 FOR I= 1TON
3110 FOR J= 1TOM
3120 LET Z(I,J)=G(I,J)
3130 NEXT J
3140 NEXT I
3150 IF C6=1. THEN 3500
3160 IF C5=1. THEN 3350
3170 LET B3=B1
3180 LET B=1.
3190 LET C5=1.
3200 LET C=2.
3210 IF B1>=A(1,1) THEN 3840
3220 LET A(1,1)=B1
3230 LET A(1,2)=B2
3240 IF B1>A(2,1) THEN 3270
3250 LET A(2,1)=B1
3260 LET A(2,2)=B2
3270 IF E=3. THEN 3290
3280 GO TO 3840
3290 FOR I= 1TON
3300 FOR J= 1TOM
3310 LET U(I,J)=G(I,J)
3320 NEXT J
3330 NEXT I
3340 GO TO 3840
3350 IF B1>=B3 THEN 3860
3360 IF B1>=A(2,1) THEN 3380
3370 LET A(2,1)=B1
3380 LET B3=B1
3390 LET C=C+1.
3400 IF B2>=A(2,2) THEN 3420
3410 LET A(2,2)=B2
3420 IF E=3. THEN 3440
3430 GO TO 3840
3440 FOR I= 1TON
3450 FOR J= 1TOM
3460 LET V(I,J)=G(I,J)
3470 NEXT J
3480 NEXT I
3490 GO TO 3840
3500 IF C5=1. THEN 3690
3510 LET B3=B1
3520 LET B=1.
3530 LET C5=1.
3540 LET C=2.
3550 IF B1>=A(3,1) THEN 3840
3560 LET A(3,1)=B1
3570 LET A(3,2)=B2
3580 IF B1>A(4,1) THEN 3610
3590 LET A(4,1)=B1
```

```
3600 LET A(4,2)=B2
3610 IF E=3. THEN 3630
3620 GO TO 3840
3630 FOR I=1TON
3640 FOR J= 1TOM
3650 LET W(I,J)=G(I,J)
3660 NEXT J
3670 NEXT I
3680 GO TO 3840
3690 IF B1>=B3 THEN 3860
3700 IF B1>=A(4,1) THEN 3720
3710 LET A(4,1)=B1
3720 LET B3=B1
3730 LET C=C+1.
3740 IF B2>=A(4,2) THEN 3760
3750 LET A(4,2)=B2
3760 IF E=3. THEN 3780
3770 GO TO 3840
3780 FOR I= 1TON
3790 FOR J= 1TOM
3800 LET Z(I,J)=G(I,J)
3810 NEXT J
3820 NEXT I
3830 GO TO 3840
3840 IF C4=1. THEN 2380
3850 GO TO 1610
3860 IF C<=A(1,3) THEN 3880
3870 LET A(1,3)=C
3880 NEXT H
3890 IF C6=1. THEN 3920
3900 LET C6=1.
3910 GO TO 1510
3920 GOTO 3990
3930 PRINT
3940 PRINT "C7="C7;"C="A(1,3)
3950 PRINT "R.S.";TAB(12);A(1,1);TAB(25);A(1,2);
3960 PRINT TAB(38);A(2,1);TAB(51);A(2,2)
3970 PRINT "L.S.";TAB(12);A(3,1);TAB(25);A(3,2);
3980 PRINT TAB(38);A(4,1);TAB(51);A(4,2)
3990 IF E=3. THEN 4010
4000 GO TO 4440
4010 PRINT
4020 PRINT "SCHEDULE-U(I,J)"
4030 GOSUB 650
4040 FOR I= 1TON
4050 PRINT USING 740;I;
4060 FOR J= 1TOM
4070 PRINT U(I,J);
4080 NEXT J
4090 PRINT
4100 NEXT I
```

```
4110 PRINT
4120 PRINT
4130 PRINT "SCHEDULE-U(I,J)"
4140 GOSUB 650
4150 FOR I= 1TON
4160 PRINT USING 740,I;
4170 FOR J= 1TOM
4180 PRINT U(I,J);
4190 NEXT J
4200 PRINT
4210 NEXT I
4220 PRINT
4230 PRINT
4240 GOSUB 650
4250 FOR I= 1TON
4260 PRINT USING 740,I;
4270 FOR J= 1TOM
4280 PRINT W(I,J);
4290 NEXT J
4300 PRINT
4310 NEXT I
4320 PRINT
4330 PRINT
4340 PRINT "SCHEDULE-Z(I,J)"
4350 GOSUB 650
4360 FOR I=1TON
4370 PRINT USING 740,I;
4380 FOR J= 1TOM
4390 PRINT Z(I,J);
4400 NEXT J
4410 PRINT
4420 NEXT I
4430 PRINT
4440 IF C7=1. THEN 4520
4450 IF C7=2. THEN 4580
4460 IF C7=3. THEN 4640
4470 IF C7=4. THEN 4700
4480 IF C7=5. THEN 4770
4490 IF C7=6. THEN 4830
4500 IF C7=7. THEN 4890
4510 IF C7=8. THEN 4950
4520 LET C7=2.
4530 LET X1=1.
4540 LET X2=1.
4550 LET X3=0.
4560 LET X4=0.
4570 GO TO 1410
4580 LET C7=3.
4590 LET X1=1.
4600 LET X2=0.
```

```
4610 LET X3=1.
4620 LET X4=0.
4630 GO TO 1410
4640 LET C7=4.
4650 LET X1=1.
4660 LET X2=0.
4670 LET X3=0.
4680 LET X4=1.
4690 GO TO 1410
4700 LET C7=5.
4710 LET L=2.
4720 LET X1=1.
4730 LET X2=0.
4740 LET X3=0.
4750 LET X4=0.
4760 GO TO 1410
4770 LET C7=6.
4780 LET X1=0.
4790 LET X2=1.
4800 LET X3=0.
4810 LET X4=0.
4820 GO TO 1410
4830 LET C7=7.
4840 LET X1=0.
4850 LET X2=0.
4860 LET X3=1.
4870 LET X4=0.
4880 GO TO 1410
4890 LET C7=8.
4900 LET X1=0.
4910 LET X2=0.
4920 LET X3=0.
4930 LET X4=1.
4940 GO TO 1410
4950 IF E=1. THEN 5090
4960 PRINT
4970 PRINT"***SCHEDULE OF JOB START TIMES***"
4980 PRINT
4990 GOSUB 650
5000 FOR I = 1 TO N
5010 PRINT USING 740,I;
5020 FOR J= 1TOM
5030 PRINT TAB (5.*J);Z(I,J);
5040 NEXT J
5050 PRINT
5060 NEXT I
5070 PRINT
5080 PRINT"TOTAL TIME =";A(2,3);" AVERAGE TIME =";A(3,3)
5090 GO TO 5660
5100 REM RIGHT SHIFT RULES WITHOUT T(I)
5110 GOSUB 5530
```

```
5120 IF J=0. THEN 2160
5130 LET I=J
5140 LET Y=M(I+1,1+D(I))
5150 FOR K=1 TO N
5160 IF K=I THEN 5210
5170 IF T(I)-P(I,Y) >= G(K,Y)+P(K,Y) THEN 5210
5180 IF T(I)<=G(K,Y) THEN 5210
5190 LET T(I)=G(K,Y)
5200 GO TO 5150
5210 NEXT K
5220 REM UPDATE
5230 LET G(J,Y)=T(I)-P(J,Y)
5240 LET F(J)=G(J,Y)
5250 LET R(J)=R(J)-P(J,Y)
5260 LET D(J)=D(J)-1.
5270 FOR I=1 TON
5280 LET T(I)=F(I)
5290 NEXT I
5300 GO TO 5110
5310 REM LEFT SHIFT RULES WITHOUT T(I)
5320 GOSUB 5530
5330 IF J=0. THEN 2940
5340 LET I=J
5350 LET Y=M(I+1,1+D(I))
5360 FOR K=1 TO N
5370 IF K=I THEN 5420
5380 IF T(I)+P(I,Y)<=G(K,Y) THEN 5420
5390 IF T(I)>=G(K,Y)+P(K,Y) THEN 5420
5400 LET T(I)=G(K,Y)+P(K,Y)
5410 GO TO 5360
5420 NEXT K
5430 REM UPDATE
5440 LET G(J,Y)=T(I)
5450 LET F(J)=G(J,Y)+P(J,Y)
5460 LET R(J)=R(J)-P(J,Y)
5470 LET D(J)=D(J)+1.
5480 FOR I=1 TO N
5490 LET T(I)=F(I)
5500 NEXT I
5510 GO TO 5320
5520 REM SUBROUTINE FIND MAX PRIORITY WITHOUT T(I)
5530 LET R=-1.
5540 LET J=0.
5550 FOR I= 1TON
5560 IF R(I)=0. THEN 5640
5570 LET Y=M(I+1,1+D(I))
5580 LET Z1=X1*P(I,Y)+X2*R(I)+X3*(1000.-P(I,Y))+X4*(4000.-R(I))
5590 IF Z1<R THEN 5640
5600 IF Z1>R THEN 5620
5610 IF P(J,M(J+1,1+D(J)))<=P(I,M(I+1,1+D(I))) THEN 5640
5620 LET R=Z1
```

```
5630 LET J=1
5640 NEXT I
5650 RETURN
5660 STOP
6000 REM THE FOLLOWING IS SAMPLE DATA AND SHOULD BE DELETED
6010 REM DUMMY N,M TO CAUSE INSTRUCTIONS TO BE PRINTED
6020 DATA 0,0
6030 REM 6 JOBS ON 6 MACHINES
6040 DATA 6,6
6050 REM SEQUENCE DATA
6060 DATA 2,3,1,4,6,5
6070 DATA 5,1,2,6,3,4
6080 DATA 4,5,1,2,6,3
6090 DATA 2,1,3,4,5,6
6100 DATA 5,2,1,6,3,4
6110 DATA 4,1,6,2,5,3
6120 REM PROCESSING TIME DATA
6130 DATA 3,6,1,7,6,3
6140 DATA 10,8,5,4,10,10
6150 DATA 9,1,5,4,7,8
6160 DATA 5,5,5,3,8,9
6170 DATA 3,3,9,1,5,4
6180 DATA 10,3,1,3,4,9
6190 REM END SAMPLE DATA
9999 END
```
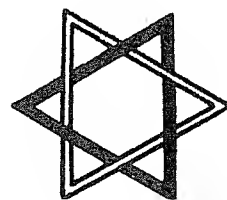
EXAMPLE

This example uses a 6/job - 6/machine problem as its entered data. While the
optimal total time for this problem is 55, please note - the best schedule
found by this program has a total time of 57.

*RUN

GEOMETRIC SIMULATOR

INSTRUCTIONS:
THIS PROGRAM CALCULATES A JOB SHOP SCHEDULE USING
A HEURISTIC GEOMETRIC METHOD.
ENTER DATA IN THE FOLLOWING ORDER, BEGINNING
IN LINE 6000:

   *   THE NUMBER OF JOBS TO BE PROCESSED,
   *   THE NUMBER OF MACHINES,
   *   THE SEQUENCE MATRIX S(I,J) BY ROWS
       WHERE S(I,J) IS THE SEQUENCE OF THE I'TH
       JOB ON THE J'TH MACHINE,
   *   THE PROCESS TIME MATRIX P(I,J) BY ROWS
       WHERE P(I,J) IS THE PROCESS TIME REQUIRED
       BY THE I'TH JOB ON THE J'TH MACHINE.

SAMPLE DATA IS ALREADY IN LINES 6000-6190.

TO EXECUTE THE PROGRAM

   ENTER YOUR DATA
   TYPE "RUN"

THIS PROGRAM REQUIRES LARGE AMOUNTS OF PROCESSOR TIME.

THE FOLLOWING IS A SAMPLE EXECUTION USING THE SAMPLE DATA.

SCHEDULING 6-JOBS ON 6-MACHINES
INPUT
SEQUENCE MATRIX-S

MACHINE	1	2	3	4	5	6
JOB						
1	2	3	1	4	6	5
2	5	1	2	6	3	4
3	4	5	1	2	6	3
4	2	1	3	4	5	6
5	5	2	1	6	3	4
6	4	1	6	2	5	3

PROCESSING TIME MATRIX-P

MACHINE	1	2	3	4	5	6
JOB
1	3	6	1	7	6	3
2	10	8	5	4	10	10
3	9	1	5	4	7	8
4	5	5	5	3	8	9
5	3	3	9	1	5	4
6	10	3	1	3	4	9

SCHEDULE OF JOB START TIMES

MACHINE	1	2	3	4	5	6
JOB
1	1	16	0	26	51	38
2	38	0	8	48	13	28
3	18	27	1	6	44	10
4	13	11	18	23	26	45
5	48	32	27	52	35	41
6	28	13	44	16	40	19

TOTAL TIME = 57   AVERAGE TIME = 52

SHIPPING:

DESCRIPTION

This solves the transportation problem. It is a special case of a scheduling assignment. It will compute the cheapest schedule of shipments between "M" factories and "N" warehouses. Where each factory has different production cost and schedules and each warehouse has different demand rates; and the cost of shipping to the various warehouses are also different.

USERS

This program is highly specialized but can be used by anyone who is concerned with the flow of resources, people or products, to various locations. This would include trucking and shipping companies, warehouse and factory business-men to name a few.

INSTRUCTIONS

The problem data must be entered before the program is run. The program is fully documented and should be listed for detailed instructions.

LIMITATIONS

Line 600 contains DIM C(20,20), line 630 contains MAT = ZER, line 640 contains MAT = CON, line 720 contains MAT = Read, and line 3270 contains MAT Print statements. Twenty five kilo Bytes of on line memory is required for storage and execution of this program. The program will store in 10K Bytes. The execution length requirements may be reduced by lowering the size of the DIM statements in lines 600 and 610.

```
110REM
115REM       ————————————————————————————————————————————————————————
120REM  DESCRIPTION--ALGORITHM TO SOLVE THE TRANSPORTATION PROBLEM
125REM
130REM  THE TRANSPORTATION PROBLEM RECEIVED ITS NAME BECAUSE IT ARISES
140REM  VERY NATURALLY IN THE CONTEXT OF DETERMINING OPTIMAL SHIPPING
150REM  PATTERNS. HOWEVER, MANY PROBLEMS HAVING NOTHING TO DO WITH
160REM  TRANSPORTATION FIT THE MATHEMATICAL MODEL FOR THE TRANSPORTA-
170REM  TION PROBLEM, AND CAN BE SOLVED BY ONE OF ITS EFFICIENT PRO-
180REM  CEDURES. TO ILLUSTRATE AN EXAMPLE OF THE TRANSPORTATION PROB-
190REM  LEM, SUPPOSE THAT M FACTORIES SUPPLY N WAREHOUSES WITH A CER-
200REM  TAIN PRODUCT. FACTORY I (I=1TOM) PRODUCES A(I) UNITS, AND WARE-
210REM  HOUSE J (J=1TON) REQUIRES B(J) UNITS. SUPPOSE THAT THE COST OF
220REM  SHIPPING FROM FACTORY I TO WAREHOUSE J IS DIRECTLY PROPORTIONAL
230REM  TO THE AMOUNT SHIPPED, AND THAT THE UNIT COST IS C(I,J). LET THE
240REM  DECISION VARIABLES, X(I,J), BE THE AMOUNT SHIPPED FROM FACTORY I
250REM  TO WAREHOUSE J. WHAT SHIPPING PATTERN (VALUES OF X(I,J)) MINIMIZES
260REM  TOTAL TRANSPORTATION COST?
270REM        *****************************************************
280REM  THE TRANSPORTATION ALGORITHM USED IN THIS PROGRAM HAS BEEN TRANS-
290REM  LATED INTO BASIC FROM ALGOL, AND CAME FROM THE COMMUNICATIONS OF
300REM  THE ACM,V9,#12,DEC 66. THE NATURE OF THE ALGORITHM IS EXPLAINED
310REM  IN AN ARTICLE BY FORD AND FULKERSON, "SOLVING THE TRANSPORTATION
320REM  PROBLEM",M.S.,1956. IN THE PROGRAM'S DOCUMENTATION, THIS ARTICLE
330REM  IS REFERRED TO AS "F+F".
340REM        *****************************************************
345REM
350REM     INSTRUCTIONS-- TYPE "RUN" AND FOLLOW INSTRUCTIONS.
390REM
395REM       ————————————————————————————————————————————————————
400REM  DATA SHOULD BE ENTERED IN THE FOLLOWING FORMAT:
410REM  10000 DATA M,N     'THE DIMENSIONS OF THE DISTRIBUTION MATRIX
420REM  10010 DATA A(I)    'FOR I=1 TO M, EACH FACTORY'S SUPPLY CAPACITY
430REM  10020 DATA B(J)    'FOR J=1 TO N, EACH WAREHOUSE'S DEMAND CAPACITY
440REM  10030 DATA C(1,J)  REM   J=1 TO N, THE FIRST ROW OF THE COST MATRIX
450REM  10040 DATA C(2,J)  REM   J=1 TO N, SECOND ROW OF THE COST MATRIX......
460REM  100X0 DATA C(M,J)  REM   J=1 TO N, THE LAST ROW OF THE COST MATRIX
500REM       ————————————————————————————————————————————————————
520REM  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
540REM  STEP 1
550REM  READ THE DIMENSIONS OF THE DISTRIBUTION MATRIX M,N, THE SUPPLY
560REM  AND DEMAND VECTORS,A(M),B(N), AND THE COST MATRIX,C(M,N).
570    PRINT "HAVE YOU ENTERED DATA BEGINNING IN LINE 10000?"
580    PRINT "IF NOT, LIST PROGRAM FOR INSTRUCTIONS"
590    PRINT
```

```
600 DIM A(20),B(20),C(20,20),X(20,20),N(20),U(20),G(20)
610 DIM V(20),T(20),R(20),S(50),L(500)
620 READ M,N
630 MAT X=ZER(M,N)
640 MAT U=CON
650 MAT T=ZER
660 FOR I=1 TO M
670 READ A(I)
675 LET A8 = A8 + A(I)
680 NEXT I
690 FOR J=1 TO N
700 READ B(J)
705  LET B8 = B8 + B(J)
710 NEXT J
714  IF A8=B8 THEN 720
716   GO TO 3350
720 MAT READ C(M,N)
730REM ----------------------------------------------------------1
740REM STEP 2
750REM SCAN EACH ROW OF THE COST MATRIX TO DISCOVER MINIMUM COST CELLS
760REM FOR EACH ROW SET H=THE MINIMUM COST CELL, AND REDUCE THE COST
770REM OF EACH CELL IN THE ROW BY H. THE MINIMUM COST CELL NOW BECOMES
780REM A ZERO COST CELL
790 FOR I=1 TO M
800 LET N(I)=(I-1)*N
810 NEXT I
820 LET C=G=0
830 FOR I=1 TO M
840 LET H=99999
850 FOR J=1 TO N
860 IF C(I,J)>=H THEN 880
870 LET H=C(I,J)
880 NEXT J
890 FOR J=1 TO N
900 LET C1=C(I,J)=C(I,J)-H
910 IF C1<>0 THEN 950
920 LET V(J)=0
930 LET N1=N(I)=N(I)+1
940 LET L(N1)=J
950 NEXT J
960 LET C=H*A(I)+C
970 NEXT I
980REM ----------------------------------------------------------2
990REM STEP 3
1000REM SCAN THOSE COLUMNS WITH NO ZERO COST CELLS TO DISCOVER THE
1010REM MINIMUM COST CELL. FOR EACH OF THESE COLUMNS, SET H=THE MIN-
1020REM IMUM COST CELL, AND REDUCE THE COST OF EACH CELL IN THE COLUMN
1030REM BY H. AGAIN THE MINIMUM COST CELL BECOMES A ZERO COST CELL.
1040 FOR J=1 TO N
1050 IF V(J)=0 THEN 1180
1060 LET H=99999
```

```
1070 FOR I=1 TO M
1080 IF C(I,J)>=H THEN 1100
1090 LET H=C(I,J)
1100 NEXT I
1110 FOR I=1 TO M
1120 LET C1=C(I,J)=C(I,J)-H
1130 IF C1<>0 THEN 1160
1140 LET N1=N(I)=N(I)+1
1150 LET L(N1)=J
1160 NEXT I
1170 LET C=H*B(J)+C
1180 NEXT J
1190REM -----------------------------------------------------------3
1200REM STEP 4
1210REM SCAN EACH ROW FOR ZERO COST CELLS. IF ROW SUPPLY IS LESS THAN
1220REM COLUMN DEMAND, FILL AS MUCH OF THE DEMAND AS POSSIBLE WITH EN-
1230REM TIRE ROW SUPPLY. IF ROW SUPPLY EQUALS OR IS GREATER THAN COLUMN
1240REM DEMAND, FILL ENTIRE DEMAND WITH AS MUCH SUPPLY AS IS NEEDED.
1250REM RECALCULATE THE RESIDUAL ROW SUPPLY AND COLUMN DEMAND. AT THIS
1260REM POINT WE HAVE ONLY ALLOCATED AVAILABLE SUPPLY TO FILL  THE ZERO
1270REM COST DEMAND CELLS IN EACH ROW.
1280 FOR I=1 TO M
1290 LET A1=A(I)
1300 LET N1=N(I)
1310 FOR U=(I-1)*M+1 TO N1
1320 IF A1=0 THEN 1440
1330 LET J=L(U)
1340 LET B1=B(J)
1350 IF B1=0 THEN 1430
1360 LET H=A1
1370 IF A1<B1 THEN 1390
1380 LET H=B1
1390 LET X(I,J)=H
1400 LET A1=A1-H
1410 LET B(J)=B1-H
1415REM    GO TO SUB "IN".
1420 GOSUB 2960
1430 NEXT U
1440 LET A(I)=A1
1450 LET G=G+A1
1460 NEXT I
1470REM -----------------------------------------------------------4
1480REM STEP 5
1490REM IF CUMULATIVE RESIDUAL SUPPLY IS EXHAUSTED, WE HAVE REACHED THE
1500REM FINAL SOLUTION TO THE PROBLEM.
1510 IF G=0 THEN 3260
1520REM -----------------------------------------------------------5
1530REM STEP 6
1540REM THE LABELING PROCESS IS BEGUN, EACH ROW WITH RESIDUAL SUPPLY IS
1550REM LABELED IN ACCORDANCE WITH "F+F'S" TREATMENT, P.27.
1560 MAT R=ZER
```

```
1570 LET K=0
1580 FOR I=1 TO M
1590 IF A(I)=0 THEN 1640
1600 LET K=K+1
1610 LET U(K)=I
1620 LET G(I)=99999
1630 GOTO 1650
1640 LET G(I)=0
1650 NEXT I
1660REM -------------------------------------------------6
1670REM STEP 7
1680REM SCAN EACH LABELED ROW LOOKING FOR ZERO COST CELLS  NOT APPEARING
1690REM IN PREVIOUSLY LABELED COLUMNS. THE COLUMNS IN WHICH SUCH ZERO
1700REM COST CELLS ARE FOUND ARE IN TURN LABELED. IF THERE IS A BREAK-
1710REM THROUGH, DISCONTINUE LABELING PROCESS AND TRANSFER TO LINE 1260,
1720REM "F+F", P.28.
1730 LET L=0
1740 FOR U=1 TO K
1750 LET I=U(U)
1760 LET N1=N(I)
1770 FOR S=(I-1)*N+1 TO N1
1780 LET J=L(S)
1785REM             ----J5----
1790 IF R(J)<>0 THEN 1840
1800 LET R(J)=I
1810 LET L=L+1
1820 LET U(L)=J
1830 IF B(J)>0 THEN 2120
1840 NEXT S
1850 NEXT U
1860REM -------------------------------------------------7
1870REM STEP 8
1880REM CONTINUES LABELING PROCESS,IF POINT OF NONBREAKTHROUGH IS REACH-
1890REM ED TRANSFER TO LINE 1540.
1900 IF L=0 THEN 2460
1910 LET K=0
1920 FOR U=1 TO L
1930 LET J=U(U)
1940 LET L1=T(J)
1950 FOR S=T(J-1)+1 TO L1
1960 LET I=S(S)
1970 IF G(I)<>0 THEN 2010
1980 LET G(I)=J
1990 LET K=K+1
2000 LET U(K)=I
2010 NEXT S
2020 NEXT U
2030 IF K=0 THEN 2460
2040 GOTO 1660
2050REM -------------------------------------------------8
2060REM STEP 9
```

```
2070REM BREAKTHROUGH PROCEEDURE OCCURS WHEN WE HAVE LABELED A COLUMN
2080REM THAT HAS RESIDUAL DEMAND>0. "F+F",P.29 EXPLAINS THE BREAKTHROUGH
2090REM PROCEEDURE. IF RESIDUAL SUPPLY IS EXHAUSTED, THEN THE SOLUTION
2100REM HAS BEEN REACHED. OTHERWISE RECOMMENCE LABELING PROCEEDURE BY
2110REM TRANSFERING TO LINE 850.
2120 LET H=B(J)
2130 LET P=J
2140 LET I=R(J)
2150 LET J=G(I)
2160 IF J<>99999 THEN 2200
2170 IF A(I)>=H THEN 2230
2180 LET H=A(I)
2190 GO TO 2230
2200 IF X(I,J)>=H THEN 2140
2210 LET H=X(I,J)
2220 GOTO 2140
2230 LET J=P
2240 LET B(J)=B(J)-H
2250 LET A(I)=A(I)-H
2260 LET G=G-H
2270REM      ---- RE1 ----
2280 LET I=R(J)
2290 LET X1=X(I,J)
2300 LET X(I,J)= X1 +H
2310 IF X1<>0 THEN 2330
2320 GOSUB 2960
2330 LET J=G(I)
2340 IF J=99999 THEN 1510
2350 LET X1=X(I,J)=X(I,J)-H
2360 IF X1<>0 THEN 2270
2370 GOSUB 3080
2380 GOTO 2270
2390REM ---------------------------------------------------------------9
2400REM STEP 10
2410REM NONBREAKTHROUGH PROCEEDURE OCCURS  WHEN WE CAN CARRY LABELING
2420REM PROCESS NO FURTHER, AND A BREAKTHROUGH HAS NOT BEEN REACHED.
2430REM "F+F",P.30-31 EXPLAINS THE NONBREAKTHROUGH PROCEEDURE. WHEN THIS
2440REM PROCEEDURE IS COMPLETED, RECOMMENCE LABELING PROCEEDURE BY TRANS-
2450REM FERING TO LINE 850.
2460 LET K=0
2470 LET L=N+1
2480 FOR J=1 TO N
2490 IF R(J)<>0 THEN 2530
2500 LET K=K+1
2510 LET U(K)=J
2520 GOTO 2550
2530 LET L=L-1
2540 LET V(L)=J
2550 NEXT J
2560 LET H=99999
2570 FOR I=1 TO M
```

```
2580 IF G(I)=0 THEN 2640
2590 FOR S=1 TO K
2600 LET J=U(S)
2610 IF C(I,J)>=H THEN 2630
2620 LET H=C(I,J)
2630 NEXT S
2640 NEXT I
2650 FOR I=1 TO M
2660 LET F=0
2670 IF G(I)=0 THEN 2690
2680 LET F=1
2690 LET N1=(I-1)*N
2700 FOR S=L TO N
2710 LET J=U(S)
2720 IF F=0 THEN 2750
2730 LET C1=C(I,J)
2740 GOTO 2760
2750 LET C1=C(I,J)=C(I,J)+H
2760 IF C1<>0 THEN 2790
2770 LET N1=N1+1
2780 LET L(N1)=J
2790 NEXT S
2800 FOR S=1 TO K
2810 LET J=U(S)
2820 IF F=0 THEN 2850
2830 LET C1=C(I,J)=C(I,J)-H
2840 GOTO 2860
2850 LET C1=C(I,J)
2860 IF C1<>0 THEN 2890
2870 LET N1=N1+1
2880 LET L(N1)=J
2890 NEXT S
2900 LET N(I)=N1
2910 NEXT I
2920 LET C=H*G+C
2930 GOTO 1520
2940REM ----------------------------------------------------------------10
2950REM STEP 11
2960REM GOSUB "IN"
2970 LET L1=T(J)
2980 FOR T=T(N) TO L1 STEP -1
2990 LET S(T+1)=S(T)
3000 NEXT T
3010 FOR T=J TO N
3020 LET T(T)=T(T)+1
3030 NEXT T
3040 LET S(L1+1)=I
3050 RETURN
3060REM ----------------------------------------------------------------11
3070REM STEP 12
3080REM    GO SUB "OUT"
```

```
3090 LET L1=T(J)
3100 FOR T=T(J-1) +1 TO L1
3110 IF S(T)<>I THEN 3140
3120 LET S=T
3130 GOTO 3150
3140 NEXT T
3150 FOR T=J TO N
3160 LET T(T)=T(T)-1
3170 NEXT T
3180 LET L1=T(N)
3190 FOR T=S TO L1
3200 LET S(T)=S(T+1)
3210 NEXT T
3220 RETURN
3230REM ----------------------------------------------------------12
3240REM STEP 13
3250REM PRINT THE SOLUTION MATRIX, AND THE TOTAL MINIMUM COST OF SOLUTION.
3260 PRINT"THE SOLUTION MATRIX ="
3270 MAT PRINTX;
3280 PRINT
3290 PRINT "THE TOTAL MINIMUM COST OF THE SOLUTION =",C
3295 STOP
3300REM ----------------------------------------------------------13
3350   IF A8 > B8 THEN 3380
3360   PRINT "TOTAL DEMAND EXCEEDS TOTAL SUPPLY."
3370   GO TO 3390
3380   PRINT "TOTAL SUPPLY EXCEEDS TOTAL DEMAND."
3390 PRINT "FOR THIS PROGRAM, THESE QUANTITIES MUST BE EQUAL."
3400 STOP
99999 END
```
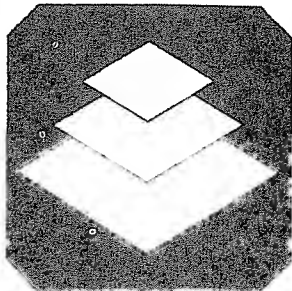
EXAMPLES

Example #1

This example shows the conventional transportation problem. Data is entered in lines 10000 thru 10050 as:

```
10000   3 factories, 5 warehouses
10010   Supply of each of the 3 factories
10020   Demand of each of the 5 warehouses
10030   Unit costs of factory #1 to ship to each warehouse
10040   Same information for factory #2
10050   Same information for factory #3
```

Example #2

In this example the transportation problem is shown with certain routes restricted as it would not be feasible or practical to ship on these routes. A high figure (99) has been entered to eliminate these routes from allocation

Example #3

This example illustrates the assignment problem, that is, the assignment of resources to jobs. The restrictions are such that each resource can only be assigned to one job and conversely each job can only have one resource. When being solved by this method a dummy resource or job must be added if the number of resources and jobs are unequal. Since this problem is looking for the maximum solution a rating matrix of some kind will generally be used. Obtaining the data for the rating matrix is probably the most difficult part of the problem. This is one of the most widely illustrated linear programming problems that can be solved by use of the transportation method.

Problem #3 has four (4) jobs but only three (3) resources, which necessitates the addition of a dummy resource in line 10000 to balance the supply and demand totals. Rating coefficients have been used in place of cost amounts in lines 10030 thru 10050. Only 3 of the 4 jobs can actually be satisified, as the dummy resource has been assigned to one of the jobs.

Example #4

This example is a representative transportation model for a production or resource scheduling problem. There are of course other problems (ex. - catering) that are similar in construction to the example problem that could also be solved by this method. These types of problems frequently have several time periods that have to be taken into consideration when solving.

In problem #4 a particular resource is available at three (3) dispatching points and the demands for those resources are at three (3) receiving points. Already determined are the demands for three periods ahead and the maximum number of resource units that can be had at each dispatching point for those time periods. As these resources are not readily available there is a time lag in getting them to the demand points and it may be necessary to retain resources at dispatch points for one or two periods in order to satisfy the demand. A holdover charge of five cost units per unit of resource per time period is contained in this problem and a slack variable has been added as a dummy destination due to this holdover charge. This problem also contains several routes which are actually meaningless as the resources available at a dispatch point are insufficient to meet the demands for a particular period and in another instance a receiving point is too far from the dispatching point to avail itself of the resources during a particular time period.

RUN 1


```
*10000   DATA   3,5
*10010   DATA   1000,500,600
*10020   DATA   400,700,300,500,500
*10030   DATA   4,6,7,4,6
*10040   DATA   7,5,8,5,8
*10050   DATA   6,4,6,7,5
*RUN
```


HAVE YOU ENTERED DATA BEGINNING IN LINE 10000?
IF NOT, LIST PROGRAM FOR INSTRUCTIONS

THE SOLUTION MATRIX =

```
   400       0       0     400     200

     0     700       0     100       0

     0       0     300       0     300
```

THE TOTAL MINIMUM COST OF THE SOLUTION =   11700

RUN 2

```
*10000    DATA   3,5

*10010    DATA   900,800,700
*10020    DATA   400,700,300,500,500
*10030    DATA   4,6,7,99,99
*10040    DATA   7,5,0,5,8
*10050    DATA   99,99,6,7,5
*RUN
```

HAVE YOU ENTERED DATA BEGINNING IN LINE 10000?
IF NOT, LIST PROGRAM FOR INSTRUCTIONS

THE SOLUTION MATRIX =

```
     400      400      100        0        0

       0      300        0      500        0

       0        0      200        0      500
```

THE TOTAL MINIMUM COST OF THE SOLUTION =   12400

RUN 3

```
READY
*10000    DATA   4,4
*10010    DATA   1,1,1,1
*10020    DATA   1,1,1,1
*10030    DATA   -90,-95,-94,-93
*10040    DATA   -92,-98,-93,-94
*10050    DATA   -91,-97,-96,-92
*10060    DATA   0,0,0,0
```

*RUN

HAVE YOU ENTERED DATA BEGINNING IN LINE 10000?
IF NOT, LIST PROGRAM FOR INSTRUCTIONS

THE SOLUTION MATRIX =

0	0	0	1
0	1	0	0
0	0	1	0
1	0	0	0

THE TOTAL MINIMUM COST OF THE SOLUTION =   -287

RUN 4


```
READY
*10000    DATA    9,10
*10010    DATA    13,20,15,8,8,8,6,6,6
*10020    DATA    4,11,6,8,10,9,12,7,4,19
*10030    DATA    2,6,99,7,11,99,12,16,19,0
*10040    DATA    3,7,99,8,12,13,13,17,18,0
*10050    DATA    99,99,5,99,99,10,99,11,15,0
*10060    DATA    99,99,99,2,6,99,7,11,99,0
*10070    DATA    99,99,99,3,7,99,8,12,13,0
*10080    DATA    99,99,99,99,99,5,99,99,8,0
*10090    DATA    99,99,99,99,99,99,2,6,99,0
*10100    DATA    99,99,99,99,99,99,3,7,99,0
*10110    DATA    99,99,99,99,99,99,99,99,5,0
```

638

HAVE YOU ENTERED DATA BEGINNING IN LINE 10000?
IF NOT, LIST PROGRAM FOR INSTRUCTIONS

THE SOLUTION MATRIX =

```
4    9    0    0    0    0    0    0    0    0
0    2    0    0    2    0    0    0    0   16
0    0    6    0    0    1    0    7    0    1
0    0    0    8    0    0    0    0    0    0
0    0    0    0    2    0    0    0    0    0
0    0    0    0    0    8    0    0    0    0
0    0    0    0    0    0    6    0    0    0
0    0    0    0    0    0    6    0    0    0
0    0    0    0    0    0    0    0    4    2
```

THE TOTAL MINIMUM COST OF THE SOLUTION =   379

STOCKS:

DESCRIPTION

This program determines the intrinsic value of stock that may be considered for a capital investment. The intrinsic value is the discounted present value of the dividends expected.

USERS

This program will be of use to individuals or companies who invest in stocks or play the stock market.

INSTRUCTIONS

During execution a full set of instructions will be listed to your terminal. It will prompt for all required data during execution. For detailed program information list Stocks.

LIMITATIONS

Line 1410 contains MAT Input, line 1450 contains NUM (X) statements. The NUM (X) statement is used throughout the program. Sample data is in lines 3910 to 3930 and should be removed before the program is run.
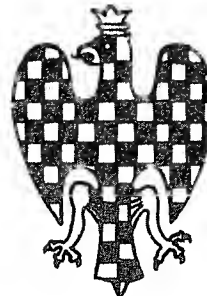
```
110REM
120REM DESCRIPTION-- THIS BASIC PROGRAM CALCULATES THE INTRINSIC
130REM              VALUE OF A STOCK BY THE MOLODOVSKY METHOD.
140REM
150REM   INSTRUCTIONS-- "RUN" THE PROGRAM AND INPUT WILL BE
160REM                  REQUESTED.
170REM                  SAMPLE DATA FOR A TRIAL RUN IS PRESENTLY
180REM                  IN THE PROGRAM......
190REM
200REM
220REM*************************************************************
230 DIM X(80),G(80),C(80),V(80),Y(80),R(80)
235 DIM T(10),Z(10),K(10)
240 PRINT "DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA";
250 INPUT Q$
260 LET R$ = "YES"
270 IF Q$="YES" THEN 410
280 IF Q$="NO" THEN 850
290 IF Q$<>"DATA" THEN 370
300REM
310 REM DATA INPUT
320 FOR I=0 TO 9
330 READ A(I)
340 NEXT I
350 GO TO 970
360REM
370 PRINT "INCORRECT RESPONSE, RETYPE IT";
380 GO TO 250
390REM
400 REM SLOW INPUT
410 PRINT "ENTER THE INFORMATION AS CALLED FOR"
420 PRINT
430 PRINT "DO YOU WANT TO DETERMINE INTRINSIC VALUE (ANSWER '1') OR THE"
440 PRINT "REQUIRED DURATION OF A SPECIFIC GROWTH RATE (ANSWER '2'), OR"
450 PRINT "THE REQUIRED FUTURE PRICE/EARNINGS RATIO (ANSWER '3')";
460 PRINT
470 PRINT TAB(40);"ITEM 0";TAB(50);
480 INPUT A(0)
490 PRINT
500 PRINT "CURRENT MARKET PRICE IN $";TAB(40);"ITEM 1";TAB(50);
510 INPUT A(1)
520 PRINT "CURRENT EARNINGS PER SHARE IN $";TAB(40);"ITEM 2";TAB(50);
530 INPUT A(2)
```

```
540 PRINT "NORMALIZED EARNINGS THIS PERIOD IN $";TAB(40);
545 PRINT "ITEM 3";TAB(50);
550 INPUT A(3)
560 PRINT"LONG TERM DISCOUNT RATE(AS A DEC.)";TAB(40);"ITEM 4";TAB(50);
570 INPUT A(4)
580 PRINT "LONG TERM DIVIDEND PAYOUT RATIO"
590 PRINT "(AS A DECIMAL)";TAB(40);"ITEM 5";TAB(50);
600 INPUT A(5)
610 PRINT
615REM        SEARCH FOR DURATION OF GR RATE
620 IF A(0)=2 THEN 740
625REM        SEARCH FOR FUTURE R/E RATIO
630 IF A(0)=3 THEN 820
640 PRINT "DO YOU WANT TO FORECAST A LONG TERM GROWTH RATE (1), OR A"
650 PRINT "PRICE/EARNINGS RATIO AT A FUTURE POINT IN TIME (2)"
660 PRINT TAB(40);"ITEM 6";TAB(50);
670 INPUT A(6)
680 PRINT
690 IF A(6)=1 THEN 740
700 PRINT "PRICE-EARNINGS RATIO AT FUTURE POINT";TAB(40);"ITEM 8";TAB(50)
710 INPUT A(8)
720 PRINT "ITEM 7,LONG TERM GROWTH RATE, IS NOT NEEDED FOR THIS CASE."
730 GO TO 770
740 PRINT"LONG TERM GROWTH RATE(AS A DECIMAL)";TAB(40);"ITEM 7";TAB(50);
750 INPUT A(7)
760 PRINT "ITEM 8, P/E RATIO AT FUTURE, IS NOT NEEDED FOR THIS CASE."
770 PRINT
780 PRINT "CURRENT YEAR (E.G. 1969)";TAB(40);"ITEM 9";TAB(50);
790 INPUT A(9)
800 PRINT
810 PRINT
820 GO TO 970
830REM
840 REM FAST INPUT
850 PRINT "ENTER ITEMS 0 THROUGH 6 AND ITEM 9";
860 INPUT A(0),A(1),A(2),A(3),A(4),A(5),A(6),A(9)
865REM        SEARCH FOR DURATION OF GROWTH RATE.
870 IF A(0)=2 THEN 930
875REM        SEARCH FOR FUTURE P/E RATIO.
880 IF A(0)=3 THEN 950
890 IF A(6)=1 THEN 930
900 PRINT "ITEM 8";
910 INPUT A(8)
920 GO TO 960
930 PRINT "ITEM 7";
940 INPUT A(7)
950 PRINT
960REM
970 LET P=A(1)
980 LET Y=A(2)
990 LET Y(0)=A(3)
1000 IF A(0)=3 THEN 1060
```

```
1010 IF A(7)<A(4) THEN 1060
1020 PRINT"LONG TERM DISCOUNT RATE MUST BE GRTR. THAN LONG TERM GROWTH"
1030 PRINT "RATE TO MAKE SOLUTION FEASIBLE; ENTER LARGER DISCOUNT RATE";
1040 INPUT A(4)
1050 GOTO 1010
1060 IF Q$<>"YES" THEN 1230
1070REM
1080 REM SLOW INPUT
1090 PRINT "      YOU WILL NOW BE ASKED TO FORECAST A GROWTH RATE"
1100 PRINT "IN EARNINGS, THE DURATION OF THAT RATE IN YEARS, THE"
1110 PRINT "DIVIDEND PAYOUT RATIO FOR THESE YEARS,DISCOUNT RATE FOR THE"
1120 PRINT "YEARS, THE TYPE OF ADJUSTMENT FROM THIS TO THE NEXT GROWTH"
1130 PRINT "RATE (ANSWER '1' FOR STRAIGHT LINE ADJUSTMENT, '2' FOR FAST"
1140 PRINT "AND '3' FOR SLOW ADJUSTMENT), THE DURATION OF THE ADJUSTMENT"
1150 PRINT "PERIOD IN YRS.,THE DIVIDEND PAYOUT RATIO FOR THE ADJUSTMENT"
1160 PRINT "PERIOD, AND THE DISCOUNT RATE FOR THE ADJUSTMENT PERIOD."
1170 PRINT "THIS WILL BE REPEATED, AND YOU MAY MAKE FORECASTS REACHING"
1180 PRINT "UP TO SEVENTY FIVE YEARS FROM THE CURRENT YEAR."
1190 PRINT "WHEN YOU HAVE PREDICTED ALL THE SPECIFIC GROWTH RATES YOU"
1200 PRINT "CARE TO, TYPE A "0" AND DO A CARRIAGE RETURN........"
1220 PRINT
1230 PRINT
1240 LET T1=1
1250 LET N=0
1260REM
1270 LET D6=A(9)+1
1280REM
1290 IF Q$<>"YES" THEN 1400
1300REM
1310 REM SLOW INPUT
1320 PRINT "FOR YEARS BEGINNING IN THE YEAR PRINTED OUT BELOW, ENTER THE"
1330 PRINT "EARNINGS GROWTH RATE (AS A DECIMAL), THE DURATION OF"
1340 PRINT "THE GROWTH RATE IN YEARS, THE DIVIDEND PAYOUT RATIO FOR"
1350 PRINT "THESE YEARS, THE DISCOUNT RATE FOR THESE YEARS, THE TYPE OF"
1360 PRINT "ADJUSTMENT, THE DURATION OF THE ADJUSTMENT IN YEARS, AND THE"
1370 PRINT "DIVIDEND PAYOUT RATIO AND DISCOUNT RATE FOR THIS ADJUSTMENT"
1380 PRINT "PERIOD."
1390 PRINT
1400 PRINT D6;"    ";
1410 MAT INPUT T
1420 IF T(1)=0 THEN 2090
1425REM          GROWTH RATE      *****
1430 LET G=T(1)
1440 LET G2=G
1450 IF NUM(X)=1 THEN 1480
1455REM    FINAL YEAR AT GROWTH RATE T(1)
1460 LET T2=T1+T(2)-1
1470 GO TO 1490
1480 LET T2=T1
1490 IF NUM(X)<3 THEN 1560
1495REM     DIVIDEND PAYOUT RATIO
```

```
1500 LET D=T(3)
1510 IF NUM(X)<4 THEN 1570
1520 FOR I=T1 TO T2
1530 LET R(I)=T(4)
1540 NEXT I
1550 GOTO 1600
1560 LET D=A(5)
1570 FOR I=T1 TO T2
1580 LET R(I)=A(4)
1590 NEXT I
1595REM      COMPUTE EARNINGS AND DIVIDENDS
1600 GOSUB 3210
1610REM
1620 IF NUM(X)<5 THEN 1650
1630 LET X9=T(5)
1640 GO TO 1660
1650 LET X9=1
1660REM
1670 IF NUM(X)<6 THEN 1700
1675REM     FIRST YR. OF NEW GROWTH RATE
1680 LET E=T2+1+T(6)
1690 GO TO 1710
1700 LET E=T2+1
1710 IF NUM(X)<7 THEN 1780
1715REM     DIVIDEND PAYOUT RATIO
1720 LET D=T(7)
1730 IF NUM(X)<8 THEN 1820
1740 FOR I=T2+1 TO E
1750 LET R(I)=T(8)
1760 NEXT I
1770 GOTO 1850
1780 LET D=A(5)
1790 IF E>T2+1 THEN 1820
1800 LET R(E)=A(4)
1810 GOTO 1850
1820 FOR I=T2+1 TO E
1830 LET R(I)=A(4)
1840 NEXT I
1850REM
1860 REM SECOND & SUBSEQUENT GROWTH RATES
1870 IF NUM(X)<8 THEN 1890
1880 GOTO 1910
1890 IF NUM(X)=1 THEN 1940
1900 GOTO 1930
1905 REM     FIRST YEAR OF NEW GROWTH RATE
1910 LET D6=D6+T(2)+T(6)
1920 GO TO 1940
1930 LET D6=D6+T(2)
1940REM      APPARRENTLY NO MORE INPUT TO COME
1950  IF NUM(X)<8 THEN 2080
1960 PRINT D6;"    ";
```

```
1970 MAT INPUT T
1980 IF T(1)=0 THEN 2090
1985REM     NEXT GROWTH RATE
1990 LET G2=T(1)
1995REM    NO ADJUSTMENT PERIOD
2000 IF E=T2+1 THEN 2020
2010 ON X9 GO TO 3290,3410,3580
2020 LET G=G2
2025REM    BEGINNING NEXT GROWTH PERIOD
2030 LET T1=E
2040 GO TO 1440
2050REM     REACHED TERMINAL DATE
2060REM
2070 REM TERMINAL CALCULATION
2080 PRINT
2085REM      SEARCH FOR DURATION OF GROWTH RATE
2090 IF A(0)=2 THEN 2580
2095REM      SEARCH FOR FUTURE P/E RATIO
2100 IF A(0)=3 THEN 2770
2105REM    P/E RATIO TO BE USED NOW
2110 IF A (6) = 2  THEN 2230
2115REM     LONG TERM GROWTH RATE
2120 LET G2=A(7)
2125REM    LONG TERM DIVIDEND PAYOUT RATIO
2130 LET D=A(5)
2140 LET N=9999
2145REM     ADJUSTMENT PERIOD
2150 IF E<>T2+1 THEN 2190
2160 LET T1=T2=E
2165REM     YEAR BY YEAR EARN. & DIVS.
2170 GOSUB 3210
2180 GOTO 2210
2190 ON X9 GO TO 3290,3410,3580
2200 LET T2=E
2210 GOSUB 3740
2220 GO TO 2240
2225REM      TERMINAL P/E RATIO
2230 GOSUB 3790
2235REM      ANNUAL INTRINSC VALUES
2240 GOSUB 3830
2250 PRINT
2260 IF R$="NO" THEN 2360
2270 PRINT A(9);"VALUES"
2280 PRINT
2290 PRINT "PRICE          ";P
2300 PRINT "EARNINGS      ";Y
2310 PRINT "P/E RATIO   ";INT((P/Y)*100+.5)/100
2320 PRINT
2330 IF A(6)=2 THEN 2360
2340 PRINT "LONG TERM GROWTH RATE";A(7)
2350 PRINT
```

```
2360 PRINT "INTRINSIC VALUE";INT(V(1)*100+.5)/100
2370 PRINT
2380 IF A(6)=1 THEN 2410
2390 LET Y1=INT(A(8)*Y(T2)*100+.5)/100
2400 PRINT "FOR YEAR";A(9)+T2;"FORECASTED P/E IS";A(8);"AND PRICE IS";Y1
2410 PRINT
2420 PRINT "DO YOU WANT A LIST OF VALUES IN FUTURE YEARS";
2430 INPUT A$
2440 PRINT
2450 IF A$="YES" THEN 2470
2460 GO TO 2820
2470 PRINT"         YEAR";"            EARNINGS";"         GROWTH";
2475 PRINT"         VALUE/EARNS.";"            INTR.VALUE"
2480 PRINT TAB(21);"PER SHARE";"            RATE";"           RATIO";
2490 PRINT TAB(61);"BEG. OF YR."
2500 LET G(1)=G(2)
2510 FOR I=1 TO T2
2520 LET D1=A(9)
2530 PRINT I+D1,INT(Y(I)*100+.5)/100,INT(G(I)*100+.5)/100,
2540 PRINT INT((V(I)/Y(I))*100+.5)/100,INT(V(I)*100+.5)/100
2550 NEXT I
2560 PRINT
2570 GO TO 2820
2580 REM SEARCH FOR DURATION OF LAST SPECIFIED GROWTH RATE
2585REM      ANNUAL INTRINSIC VALUES
2590 GOSUB 3830
2600 LET G3=A(7)
2610 LET X(T2+1)=X(T2)*(1+G2)
2620 LET V(1)=V(1)+X(T2+1)/(1+A(4))↑(T2+1)
2630 LET W=V(1)+(X(T2+1)*((1+G3)/(A(4)-G3)))/(1+A(4))↑(T2+1)
2640 IF W>=P THEN 2740
2650 LET T2=T2+1
2660 IF T2>75 THEN 2700
2670 GO TO 2610
2680REM
2690 PRINT
2700PRINT"GROWTH RATE"G2;"MUST LAST BEYOND";A(9)+T2+1;"TOJUSTIFY PRICE."
2710 PRINT
2720 PRINT
2730 GO TO 2760
2740 PRINT "GROWTH RATE";G2;"MUST LAST TO";A(9)+T2+1;"TO JUSTIFY PRICE"
2750 PRINT
2760 GO TO 2820
2765REM      ANUAL INTRINSIC VALUES
2770 GOSUB 3830
2780 LET W=(P-V(1))*(1+R(T2))↑(T2)/Y(T2)
2790 PRINT
2800 PRINT "REQUIRED P/E RATIO IN";A(9)+T2;"TO JUSTIFY PRICE IS";
2805 PRINT INT(W*100+.5)/100
2810 PRINT
2820 PRINT "DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9";
```

```
2830 INPUT R$
2840 IF R$="NO" THEN 3160
2850 IF R$="STOP" THEN 3190
2860 IF R$="END" THEN 3190
2870 LET Q$="NO"
2880 PRINT "WHICH ITEMS";
2890 MAT INPUT Z
2895REM      ***     NO CHANGES    **
2900 IF NUM(X)=0 THEN 3010
2910 LET Q2=NUM(X)
2915REM        **       INDICATORS      **
2920 LET Z1=Z2=0
2930 PRINT "WHAT ARE THE NEW VALUES";
2940 MAT INPUT K
2945REM        ** NO CHANGES    **
2950 IF NUM(X)=0 THEN 3060
2960 FOR I=1 TO Q2
2970 LETA(Z(I))=K(I)
2980 IF Z(I)=0 THEN 3030
2990 IF Z(I)=6 THEN 3010
3000 GO TO 3040
3005REM    CHANGE IN A(6) MAY REQUIRE OTHER CHANGES
3010 LET Z1=1
3020 GO TO 3040
3025REM     CHANGE IN A(0) MAY REQUIRE OTHER CHANGES
3030 LET Z2=1
3040 NEXT I
3050 GO TO 3080
3060 PRINT "NO CHANGES MADE"
3070 GO TO 3150
3080 IF Z2<>1 THEN 3140
3090 IF Z1=1 THEN 3130
3100 PRINT
3110 PRINT "ITEM 6";
3120 INPUT A(6)
3130 GOTO 870
3140 IF Z1=1 THEN 890
3150 GO TO 960
3160 IF Q$<>"YES" THEN 3180
3170 LET Q$="NO"
3180 GO TO 960
3190 STOP
3200 REM YEAR BY YEAR EARNINGS & DIVS
3210 FOR J=T1 TO T2
3220 LET Y(J)=Y(J-1)*(1+G)
3230 LET G(J)=G
3240 LET X(J)=Y(J)*D
3250 NEXT J
3260 RETURN
3270REM
3280 REM STR LINE ADJ. IN GROWTH
```

```
3290 LET K1=(G-G2)/(E-T2)
3300 FOR J=T2+1 TO E
3310 LET G=G-K1
3320 LET Y(J)=Y(J-1)*(1+G)
3330 LET X(J)=Y(J)*D
3340 LET G(J)=G
3350 NEXT J
3360 LET G=G(J)=G2
3370 IF N=9999 THEN 2200
3380 GOTO 2020
3390REM
3400 REM FAST ADJ.
3410 LET K3=0
3420 FOR K=T2+1 TO E
3430 LET K3 = K3+(K-T2)
3440 NEXT K
3450 LET K1=G-G2
3460 FOR J=T2+1 TO E
3470 LET K2=E-J+1
3480 LET G=G-K1*(K2/K3)
3490 LET Y(J)=Y(J-1)*(1+G)
3500 LET X(J)=Y(J)*D
3510 LET G(J)=G
3520 NEXT J
3530 LET G=G(J)=G2
3540 IF N=9999 THEN 2200
3550 GOTO 2020
3560REM
3570 REM SLOW ADJ.
3580 LET K3=0
3590 FOR K=T2+1 TO E
3600 LET K3 = K3+(K-T2)
3610 NEXT K
3620 LET K1=G-G2
3630 FOR J=T2 +1 TO E
3640 LET K2 = J-T2
3650 LET G=G-K1*(K2/K3)
3660 LET Y(J)=Y(J-1)*(1+G)
3670 LET X(J)=Y(J)*D
3680 LET G(J)=G
3690 NEXT J
3700 LET G=G(J)=G2
3710 IF N=9999 THEN 2200
3720 GOTO 2020
3730REM
3740 REM VALUE OF LONG TERM GROWTH RATE
3750 LET X(T2)=X(T2)+(X(T2)*(1+A(7))/(A(4)-A(7)))
3760 RETURN
3770REM
3780 REM VALUE OF TERMINAL P/E RATIO
3790 LET X(T2)=X(T2)+A(8)*Y(T2)
```

```
3800 RETURN
3810REM
3820 REM ANNUAL INTR. VALUES
3830 FOR J=1 TO T2
3840    LETV(J) = 0
3850    FOR I = J TO T2
3860 LET V(J)=V(J)+X(I)/(1+R(I))↑(I-J+1)
3870    NEXT I
3880 NEXT J
3890 RETURN
3900REM
3910 DATA 1
3920 DATA 302,6.85,7,.16,.4
3930 DATA 1,.05,30,1973
3940 END
```
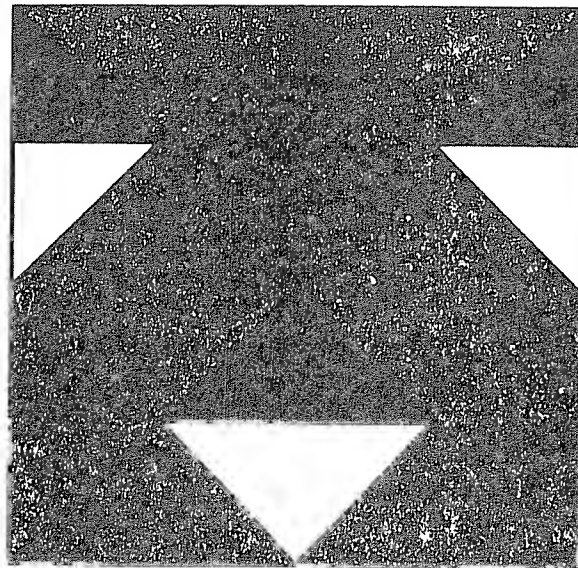
EXAMPLES

Example #1

In the example shown below the user has answered "YES" to the program's query "Do You Want Instructions for Supplying Data?". The user's response to item Ø indicates that an intrinsic value is to be calculated. Data has been supplied for items 1 thru 5 and item 6, which gives the user a choice between forecasting long-term growth rates or future price/earnings ratios has been answered by his selection of the former. The growth rate is then supplied for item 7 and the current year is entered in response to item 9; item 8 is not used due to the selection of long-term growth rate in item 6.

Now the user must supply estimates of the growth rate in earnings per share, the duration of that rate (yrs.), the dividend payout ratio for those years, the discount rate for those years, the type of adjustment to the next growth rate, the duration of the adjustment period (yrs.), the dividend payout ratio for the adjustment period, and the discount rate for the adjustment period. Note: If you wish to shift from one growth rate to another without any period in between, then show the duration of the adjustment period as a Ø.

In this example the user has forecast a growth rate in earnings per share of 25% a year for 10 years (1974-1983) with a dividend payout of 35% and a discount rate of 20%, followed by a Slow adjustment over a 5 year (1984-1988) period using the same dividend payout ratio and discount rate. The user then forecast a second set of estimates, predicting a 22% rate of growth for 5 years (1989-1993) with a dividend payout of 40% and a discount rate of 18%, followed once again by a Slow adjustment for a 2 year (1994-1995) period and the same dividend payout of 40% and discount rate of 18%. As no further data has been supplied by the user the rates beginning in 1996 will be the same as those previously entered in items 4, 5, and 7.

The output of this program summarizes some of the data supplied and prints out the intrinsic value of the stock. The user also has the option of having the computer print out a table of values in future years. Finally, the computer will offer the user a chance to change the data in any of the items Ø thru 9 and/or to modify or substitute data in the growth forecasts. A modified set of forecasts appears after the values in future years listing.

RUN 1

DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA ?YES
ENTER THE INFORMATION AS CALLED FOR

DO YOU WANT TO DETERMINE INTRINSIC VALUE (ANSWER "1") OR THE
REQUIRED DURATION OF A SPECIFIC GROWTH RATE (ANSWER "2"), OR
THE REQUIRED FUTURE PRICE/EARNINGS RATIO (ANSWER "3#)
                                            ITEM 0   ?1

CURRENT MARKET PRICE IN $ ITEM 1   ?32
CURRENT EARNINGS PER SHARE IN $ ITEM 2   ?6.85
NORMALIZED EARNINGS THIS PERIOD IN $ ITEM 3   ?7.00
LONG TERM DISCOUNT RATE(AS A DEC.)   ITEM 4   ?.16
LONG TERM DIVIDEND PAYOUT RATIO   ITEM 5   ?.4

DO YOU WANT TO FORECAST A LONG TERM GROWTH RATE (1), OR A
PRICE/EARNINGS RATIO AT A FUTURE POINT IN TIME (2)
                                            ITEM 6   ?1

LONG TERM GROWTH RATE (AS A DECIMAL)   ITEM 7   ?.05
ITEM 8, P/E RATIO AT FUTURE, IS NOT NEEDED FOR THIS CASE.

CURRENT YEAR (E.G. 1969)   ITEM 9   ?1973

YOU WILL NOW BE ASKED TO FORECAST A GROWTH RATE IN EARNINGS,
THE DURATION OF THAT RATE IN YEARS, THE DIVIDEND PAYOUT RATIO
FOR THESE YEARS, DISCOUNT RATE FOR THE YEARS, THE TYPE OF
ADJUSTMENT FROM THIS TO THE NEXT GROWTH RATE (ANSWER "1" FOR
STRAIGHT LINE ADJUSTMENT, "2" FOR FAST AND "3" FOR SLOW ADJ.),
THE DURATION OF THE ADJUSTMENT PERIOD IN YRS., THE DIVIDEND
PAYOUT RATIO FOR THE ADJUSTMENT PERIOD, AND THE DISCOUNT
RATE FOR THE ADJUSTMENT PERIOD. THIS WILL BE REPEATED, AND
YOU MAY MAKE FORECASTS REACHING UP TO SEVENTY FIVE YEARS
FROM THE CURRENT YEAR. WHEN YOU HAVE PREDICTED ALL THE
SPECIFIC GROWTH RATES YOU CARE TO, TYPE A   0 AND DO A
CARRIAGE RETURN. . . . .

Example #2

In this second example the user has called for a calculation of the required
duration of a forecasted growth rate in order to make the intrinsic value the
same as the current price, accordingly the user has changed the response to
item 0 to a 2. In addition the user has changed item 4 to 15% and item 5 to
50%, all other forecasted data remains the same as that shown in example #1.

The results of this calculation mean that in order for the intrinsic value
to equal the current market price, the growth rate predicted in the first
forecast must continue until 1991 and the long-term growth rate thereafter.
In the case of the second growth forecast the rate must continue until 1989
in order to make the intrinsic value equal to the market price.

```
FOR YEARS BEGINNING IN THE YEAR PRINTED OUT BELOW, ENTER THE
EARNINGS GROWTH RATE (AS A DECIMAL), THE DURATION OF THE
GROWTH RATE IN YEARS, THE DIVIDEND PAYOUT RATIO FOR THESE
YEARS, THE DISCOUNT RATE FOR THESE YEARS, THE TYPE OF ADJ.,
THE DURATION OF THE ADJUSTMENT IN YEARS, AND THE DIVIDEND
PAYOUT RATIO AND DISCOUNT RATE FOR THIS ADJUSTMENT PERIOD.

    1974      ?.25,10,.35,.2,3,5,.35,.2
    1989      ?.22,5,.4,.18,3,2,.4,.18
    1996      ?0

    1973 VALUES

PRICE        302
EARNINGS       6.85
P/E RATIO     44.09

LONG TERM GROWTH RATE    .05

INTRINSIC VALUE    170.4


DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?YES
```

652

YEAR	EARNINGS PER SHARE	GROWTH RATE	VALUE/EARNS. RATIO	INTR. VAL. BEG. OF YR.
1974	8.75	.25	19.47	170.4
1975	10.94	.25	18.2	199.04
1976	13.67	.25	16.98	232.21
1977	17.09	.25	15.83	270.55
1978	21.36	.25	14.73	314.77
1979	26.7	.25	13.69	365.63
1980	33.38	.25	12.7	423.96
1981	41.72	.25	11.76	490.65
1982	52.15	.25	10.86	566.59
1983	65.19	.25	10.01	652.71
1984	81.4	.25	9.21	749.88
1985	101.4	.25	8.47	858.41
1986	125.88	.24	7.79	980.5
1987	155.55	.24	7.17	1115.19
1988	191.1	.23	6.61	1263.32
1989	233.15	.22	6.11	1424.95
1990	284.44	.22	5.58	1588.18
1991	347.02	.22	5.07	1760.28
1992	423.36	.22	4.58	1938.32
1993	516.5	.22	4.1	2117.49
1994	615.49	.19	3.72	2292.49
1995	699.59	.13	3.52	2450.94
1996	733.52	.05	3.57	2622.12

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?NO

```
    1974      ?.25,10,.3,.2,3,5,.3,.2
    1989      ?.22,5,.35,.10,3,5,.35,.13
    1999      ?.10,5,.35,.16,3,5,.35,.16
    2009      ?0
```

INTRINSIC VALUE   360.22

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?NO

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?YES
WHICH ITEMS ?4,5
WHAT ARE THE NEW VALUES ?.15,.5

```
    1974      ?.25,10,.5,.15,3,5,.5,.15
    1989      ?.22,5,.5,.15,3,2,.5,.15
    1996      ?0
```

    1973 VALUES

```
PRICE        302
EARNINGS      6.85
P/E RATIO     44.09
```

LONG TERM GROWTH RATE   .05

INTRINSIC VALUE   380.43

Example #3

In the third example the user has changed his response to item 0 back to a 1 (intrinsic value calculation) and has changed his response to item 6 to call for a forecast of a price/earnings ratio. Item 7 is not used and a ratio forecast of 30 is entered in response to the initial use of item 8. As the growth forecast extends to 2013 the price/earnings ratio will be applied as of 2013. The print out shows the intrinsic value and the actual forecasted price in 2013, based on the ratio and the earnings growth projections. Note: As no succeeding growth rate was used in this particular problem, no adjustment to the next rate was specified.

Next, the user changes the ratio forecast in item 8 to 25, provides a new set of growth forecasts and obtains a new print out of the intrinsic value and price as of the year 1993. Finally the user changes item 0 to call for a calculation of the price/earnings ratio required to make the intrinsic value equal to the current market price.

```
     1973 VALUES

PRICE          302
EARNINGS        6.85
P/E RATIO      44.09

INTRINSIC VALUE    1027.44

FOR YEAR 2013 FORECASTED P/E IS    30  AND PRICE IS        188126.3

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?NO

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?YES
WHICH ITEMS ?8
WHAT ARE THE NEW VALUES ?25

    1974     ?.18,.5,.5,.15,.5,.5,.15
    1994     ?.15,10,.5,.15
```

```
       1973 VALUES

PRICE        302
EARNINGS       6.85
P/E RATIO     44.09

INTRINSIC VALUE      300.4

FOR YEAR 1993 FORECASTED P/E IS    .25   AND PRICE IS       3550.71

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?NO

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?YES
WHICH ITEMS ?0
WHAT ARE THE NEW VALUES ?3

ITEM 6 ?2

    1974     ?.18,5,.5,.15,3,5,.5,.15
    1984     ?.15,5,.5,.15

REQUIRED P/E RATIO IN 1998 TO JUSTIFY PRICE IS    27.68
```

Example #4

Example #4 uses the data from the last example and shifts the response to
item Ø back to a 1 to generate a table of future values utilizing the price/
earnings ratio calculated in the previous example.

```
       DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?YES
       WHICH ITEMS ?0
       WHAT ARE THE NEW VALUES ?1

       ITEM 6  ?2
       ITEM 8  ?27.68
```

```
1974       ?.18,5,.5,.15,3,5,.5,.15
1984       ?.15,5,.5,.15

1973 VALUES

PRICE         302
EARNINGS      6.85
P/E RATIO    44.09

INTRINSIC VALUE    301.96

FOR YEAR 1988 FORECASTED P/E IS    27.68   AND PRICE IS     1954.57

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?YES

    YEAR    EARNINGS      GROWTH      VALUE/EARNS.      INTR. VAL.
            PER SHARE     RATE          RATIO          BEG. OF YR.
    1974      8.26         .18          36.56            301.96
    1975      9.75         .18          35.2             343.13
    1976     11.5          .18          33.89            389.73
    1977     13.57         .18          32.6             442.43
    1978     16.01         .18          31.35            502.01
    1979     18.87         .18          30.16            569.31
    1980     22.19         .18          29.08            645.27
    1981     25.99         .17          28.12            730.96
    1982     30.3          .17          27.31            827.61
    1983     35.11         .16          26.68            936.6
    1984     40.37         .15          26.24           1059.54
    1985     46.43         .15          25.81           1198.28
    1986     53.39         .15          25.37           1354.81
    1987     61.4          .15          24.94           1531.33
    1988     70.61         .15          24.5            1730.33

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?END
```

Example #5

In the previous examples the user had originally responded "YES" to the initial question "Do You Want Instructions for Supplying Data?". This example demonstrates the rapid input of data when the first question is answered by a "NO".

```
DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA ?NO
ENTER ITEMS 0 THROUGH 6 AND ITEM 9   ?1,302,6.95,7,.16,.4,1,1973
ITEM 7  ?.05

     1974     ?.25,10,.35,.16,3,5,.35,.16
     1989     ?.22,5,.4,.16,3,2,.4,.16
     1996     ?0

     1973 VALUES

PRICE        302
EARNINGS       6.95
P/E RATIO     44.09

LONG TERM GROWTH RATE     .05

INTRINSIC VALUE     241.34

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?NO

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?END
```

Example #6

This example shows an alternate form of input utilizing DATA lines for entry.
All data listed is presently in the program and the user may overlay it with
his own data.

```
READY
*LIST 3910-3930

3910 DATA 1
3920 DATA 302,6.85,7,.16,.4
3930 DATA 1,.05,30,1973

READY

*RUN


DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA ?DATA

     1974      ?.25,10,.4,.16,3,5,.4,.16
     1989      ?.22,5,.4,.16,3,2,.4,.16
     1996      ?0

     1973 VALUES

PRICE       302
EARNINGS      6.85
P/E RATIO    44.09

LONG TERM GROWTH RATE    .05

INTRINSIC VALUE    251.31

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?NO

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?NO

     1974      ?END
```

DO YOU WANT A LIST OF VALUES IN FUTURE YEARS ?NO

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?YES
WHICH ITEMS ?0
WHAT ARE THE NEW VALUES ?2

ITEM 6 ?1
ITEM 7 ?.05

    1974   ?.25

GROWTH RATE   .25 MUST LAST TO 1991 TO JUSTIFY PRICE

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?NO

    1974   ?.25,10,.5,.15,3,5,.5,.15
    1989   ?.22

GROWTH RATE   .22 MUST LAST TO 1992 TO JUSTIFY PRICE

DO YOU WANT TO CHANGE ANY OF ITEMS 0 TO 9 ?YES
WHICH ITEMS ?0,6
WHAT ARE THE NEW VALUES ?1,2
ITEM 8 ?30

    1974   ?.22,10,.5,.14,3,5,.5,.15
    1989   ?.18,10,.5,.15,3,5,.5,.15
    2004   ?.15,10,.5,.15

SWITCH:

<u>DESCRIPTION</u>

Switch computes the effects of a Bond switch. A bond switch is when an investor sells a bond and purchases another. The reason an investor may do this is because he expects that a change in the yield will make the switch financially advantageous.

<u>USERS</u>

Persons or companies who invest in Bonds will find this program may save considerable money on some of their transactions

<u>INSTRUCTIONS</u>

The data may be entered in data statements before program execution or it may be entered during execution through program prompting. The program should be listed for a complete set of instructions detailing data entry and format.

<u>LIMITATIONS</u>

Line 4140 contains a ON___GOTO statement and line 4560 contains an ABS( ) statement. Program lines 1110 to 1160 contain sample data and should be removed before running the program. The source code for Switch will require 20K Bytes of memory for storage. It will store and execute in 25K Bytes of memory space.

```
15REM
20REM   THIS BASIC PROGRAM IS TO CALCULATE AND ANALYZE THE
25REM   EFFECT OF A BOND SWITCH
26REM
27REM      *******************************************
36REM   TYPE "RUN" TO USE.   DATA INPUTS ARE DESCRIBED IN
37REM   PRINT STATEMENTS.
50REM
55REM * * * * * * * * * * * * * * * * * * * * * * * * * *
60REM
110REM  DESCRIPTION OF VARIABLES IN DATA
120REM C1,C2: ANNUAL COUPON ON OLD AND NEW BONDS
130REM B1: BOOK VALUE OLD BOND
140REM P1,P2: PRICE OLD AND NEW BONDS
150REM M1,M3,M2,M4: MATURITIES OLD AND NEW BONDS IN YEARS AND MONTHS
160REM R1,R2: REDEMPTION PRICE OLD AND NEW BONDS
170REM T1,T2: TAX RATE ON THE INTEREST OF OLD AND NEW BONDS
180REM S1,S2: TERMINAL DATE IN YEARS AND MONTHS HENCE
190REM Y3,Y4: ESTIMATED YIELD ON OLD AND NEW BONDS AT TERMINAL DATE
200REM T3,T5: CAPITAL LOSS RATE NOW AND AT TERMINAL DATE
210REM T4,T6: CAPITAL GAIN RATE NOW AND AT TERMINAL DATE
220REM D3,D4: CAPITAL GAIN RATE AT MATURITY OF OLD AND NEW BOND
230REM Y9: AFTER TAX REINVESTMENT RATE FOR COUPONS
240REM
250 PRINT "DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA";
260 INPUT A$
270 IF A$="DATA" THEN 1020
280 IF A$="NO" THEN 910
290 PRINT
300 PRINT "ENTER THE FOLLOWING ITEMS OF DATA"
310 PRINT
320 PRINT "ITEM 1";TAB(12);"BOOK VALUE OF OLD BOND";TAB(45);
330 INPUT B1
340 PRINT
350 PRINT "ITEM 2";TAB(12);"PRICE OF OLD BOND";TAB(45);
360 INPUT P1
370 PRINT
380 PRINT "ITEM 3";TAB(12);"COUPON ON OLD BOND";TAB(45);
390 INPUT C1
400 PRINT
410 PRINT "ITEMS 4 & 5";TAB(12);"MATURITY OLD BOND IN YRS,MONTHS";TAB(45);
420 INPUT M1,M3
430 PRINT
```

```
440 PRINT "ITEM 6";TAB(12);"PROCEEDS ON REDEMPTION OLD BOND";TAB(45);
450 INPUT R1
460 PRINT
470 PRINT "ITEM 7";TAB(12);"TAX RATE ON OLD BOND INTEREST PMTS";TAB(45);
480 INPUT T1
490 PRINT
500 PRINT "ITEM 8";TAB(12);"PRICE OF NEW BOND";TAB(45);
510 INPUT P2
520 PRINT
530 PRINT "ITEM 9";TAB(12);"COUPON ON NEW BOND";TAB(45);
540 INPUT C2
550 PRINT
560 PRINT "ITEMS 10,11";TAB(12);"MATURITY NEW BOND- YRS,MONTHS";TAB(45);
570 INPUT M2,M4
580 PRINT
590 PRINT "ITEM 12";TAB(12);"PROCEEDS ON REDEMPTION NEW BOND";TAB(45);
600 INPUT R2
610 PRINT
620 PRINT "ITEM 13";TAB(12);"TAX RATE ON NEW BOND INTEREST PMTS";TAB(45);
630 INPUT T2
640 PRINT
650 PRINT "ITEMS 14,15";TAB(12);"YRS,MONTHS TO TERMINAL DATE";TAB(45);
660 INPUT S1,S2
670 PRINT
680 PRINT "ITEM 16";TAB(12);"PREDICT YIELD OLD BOND TERM. DATE";TAB(45);
690 INPUT Y3
700 PRINT
710 PRINT "ITEM 17";TAB(12);"PREDICT YIELD NEW BOND TERM. DATE";TAB(45);
720 INPUT Y4
730 PRINT
740 PRINT "ITEMS 18,19";TAB(12);"CAP LOSS,GAIN RATES NOW";TAB(45);
750 INPUT T3,T4
760 PRINT
770 PRINT "ITEMS 20,21";TAB(12);"CAP LOSS, GAIN RATES TERM.DATE";TAB(45);
780 INPUT T5,T6
790 PRINT
800 PRINT "ITEM 22";TAB(12);"CAP GAIN RATE AT MATURITY OLD BOND";TAB(45);
810 INPUT D3
820 PRINT
830 PRINT "ITEM 23";TAB(12);"CAP GAIN RATE AT MATURITY NEW BOND";TAB(45);
840 INPUT D4
850 PRINT
860 PRINT "ITEM 24";TAB(12);"AFTER TAX REINVEST RATE COUPONS";TAB(45);
870 INPUT Y9
880 PRINT
890 GO TO 1170
900 PRINT
910 PRINT "ENTER DATA ITEMS 1 THROUGH 13"
920 PRINT
930 INPUT B1,P1,C1,M1,M3,R1,T1,P2,C2,M2,M4,R2,T2
940 PRINT
```

```
950 PRINT "ENTER DATA ITEMS 14 THROUGH 24"
960 PRINT
970 INPUT S1,S2,Y3,Y4,T3,T4,T5,T6,D3,D4,Y9
980 PRINT
990 GO TO 1170
1000REM
1010REM DATA ON OLD BOND
1020 READ B1,P1,C1,M1,M3,R1,T1
1030REM
1040REM DATA ON NEW BOND
1050 READ    P2,C2,M2,M4,R2,T2
1060REM
1070 READ S1,S2,Y3,Y4
1080 READ T3,T4,T5,T6
1090 READ D3,D4
1100 READ Y9
1110 DATA 90.375,91,2.75,2,5,100,.48
1120 DATA 84,5.125,32,9,100,.48
1130 DATA 2,0,.05,.045
1140 DATA .25,.25,.25,.25
1150 DATA .25,.25
1160 DATA .03
1170REM SEMI-ANNUAL COUPON
1180 DIM A(50),L(50,50)
1190 LET C1 = C1/2
1200 LET C2 = C2/2
1210 REm
1220 REm YIELD ON OLD BOND
1230 REm
1235REM   SEMI-ANNUAL COUPONS TO MATURITY OLD BOND
1240 LET M=M1*2+M3/6
1250 LET M7 = 0
1260 LET M8 = INT(M)-1
1265REM    FRACTION OF PERIOD UNTIL NEXT COUPON
1270 LET M9=M-INT(M)
1280 LET C = C1
1285REM   MARKET VALUE OLD BOND
1290 LET U1=P1+C1*(1-M9)
1295REM   MARKET VALUE AT MATURITY
1300 LET U2=R1+C1
1310 GOSUB 4530
1315REM   YIELD TO MATURITY
1320 LET Q1=2*(EXP(Y)-1)
1330 REm
1340 REm AFTER TAX YIELD ON OLD BOND
1350 REm
1355REM   IS BOND SELLING AT A PREMIUM
1360 IF B1<100 THEN 1390
1365REM   PER PERIOD AMORTIZATION
1370 LET D1=(B1-100)/INT(M+1)
1380 GO TO 1400
```

```
1390 LET D1=0
1395REM    AFTER TAX AND AMORTIZATION COUPON
1400 LET C3=C1-(C1-D1)*T1
1405REM    BOOK VALUE AT MATURITY
1410 LET B9=B1-D1*(INT(M+1))
1415REM     TAX SAVINGS ON CAPITAL LOSS
1420 LET G5=D3*(R1-B9)
1430 LET C=C3
1435REM   AFTER TAX VALUE AT PRESENT
1440 LET U1=P1+C1*(1-M9)*(1-T1)
1445REM   AFTER TAX VALUE AT MATURITY
1450 LET U2=R1-G5+C3
1460 GOSUB 4530
1465REM   AFTER TAX YIELD TO MATURITY
1470 LET Q3=2*(EXP(Y)-1)
1480 REM
1490 REM AFTER TAX YIELD ON REPURCHASED OLD BOND
1500 REM
1505REM   IS BOND SELLING AT A PREMIUM
1510 IF P1<100 THEN 1540
1515REM   PER PERIOD AMORTIZATION
1520 LET D5=(P1-100)/INT(M+1)
1530 GO TO 1550
1540 LET D5=0
1545REM   AFTER TAX AND AMORTIZATION COUPON
1550 LET C5=C1-(C1-D5)*T1
1555REM   BOOK VALUE AT MATURITY
1560 LET B9=P1-D5*(INT(M+1))
1565REM   TAX SAVINGS ON CAPITAL LOSS
1570 LET G7=D3*(R1-B9)
1575REM   AFTER TAX AND AMORTIZATION COUPON
1580 LET C=C5
1585REM   AFTER TAX MARKET VALUE PRESENT
1590 LET U1=P1+C1*(1-M9)*(1-T1)
1595REM   AFTER TAX VALUE AT MATURITY
1600 LET U2=R1-G7+C5
1610 GOSUB 4530
1615REM AFTER TAX YIELD TO MATURITY
1620 LET Q5=2*(EXP(Y)-1)
1630 REM
1640 REM YIELD ON NEW BOND
1650 REM
1655REM   SEMI-ANNUAL COUPONS TO MATURITY NEW BOND
1660 LET M=M2*2+M4/6
1670 LET M7 = 0
1675REM   SEMI-ANNUAL COUPONS TO MATURITY LESS FINAL COUPON
1680 LET M8=INT(M)-1
1685REM    FRACTION OF PERIOD TO NEXT COUPON
1690 LET M9=M-INT(M)
1695REM   SEMI-ANNUAL COUPON
1700 LET C=C2
```

```
1705REM   PRESENT MARKET VALUE
1710 LET U1=P2+C2*(1-M9)
1715REM MARKET VALUE AT MATURITY
1720 LET U2=R2+C2
1730 GOSUB 4530
1735REM YIELD TO MATURITY NEW BOND
1740 LET Q2=2*(EXP(Y)-1)
1750 REm
1760 REm AFTER TAX YIELD ON NEW BOND
1770 REm
1775REM IS NEW BOND SELLING AT A PREMIUM
1780 IF P2<100 THEN 1810
1785REM   AMORTIZATION OF PREMIUM PER PERIOD
1790 LET D2=(P2-100)/INT(M+1)
1800 GO TO 1820
1810 LET D2 = 0
1815REM   AFTER TAX AND AMORTIZATION COUPON
1820 LET C4=C2-(C2-D2)*T2
1825REM    BOOK VALUE AT MATURITY
1830 LET B9=P2-D2*(INT(M+1))
1835REM    TAX SAVINGS ON CAPITAL LOSS
1840 LET G6=D4*(R2-B9)
1845REM   AFTER TAX AND AMORTIZATION COUPON
1850 LET C=C4
1855REM   AFTER TAX MARKET VALUE   PRESENT
1860 LET U1=P2+C2*(1-M9)*(1-T2)
1865REM   AFTER TAX MARKET VALUE AT MATURITY
1870 LET U2=R2-G6+C4
1880 GOSUB 4530
1885REM    AFTER TAX YIELD TO MATURITY  NEW BOND
1890 LET Q4=2*(EXP(Y)-1)
1900 PRINT "TABLE OF YIELDS AND TERMINAL VALUES AT TERM. DATE TYPE:"
1910 PRINT "    0-FOR NO TABLES"
1920 PRINT "    1-FOR SENSITIVITY OF YIELD AND TERMINAL VALUES TO TERM."
1930 PRINT "    2-FOR SENSITIVITY TO YIELD SPREAD"
1940 PRINT "    3-FOR SENSITIVITY TO YIELDS"
1950 PRINT "    4-FOR YIELDS AND TERMINAL VALUES AT YOUR TERM. DATE"
1960 PRINT
1970 PRINT "WHICH TABLE DO YOU WANT";
1980 INPUT Z
1990 PRINT
2000 LET K2=Y3
2010 LET K1=Y4
2020 LET Z1=0
2030 ON Z+1 GO TO 5190,2100,2140,2270,2390
2080 REM K,Z1 SERVE AS COUNTERS TO SET UP SENSITIVITY MATRICIES
2090 GO TO 2390
2095REM     LOOP USED TO GENERATE TIME SENSITIVITIES
2100 FOR S=1 TO S1*2+S2/6+2
2110 LET Z1=Z1+1
2120 LET A(Z1)=S/2
```

```
2130 GO TO 2400
2140 LET   Y6=Y4-Y3
2150 PRINT "DESIGNATE RANGE OF SPREAD(NEW BOND YIELD-OLD BOND YIELD AT"
2160 PRINT "TERMINAL DATE) INPUT HIGH SPREAD, LOW SPREAD AND INCREMENT"
2170 PRINT "IN BASIS POINTS.  FOR EXAMPLE, -100,100,25."
2180 INPUT Z4,Z5,Z6
2190 LET Z4=Z4/10000
2200 LET   Z5=Z5/10000
2210 LET Z6=Z6/10000
2220 FOR K1=Y3+Z4 TO Y3+Z5 STEP Z6
2230   LET Z1=Z1+1
2240 LET A(Z1)=(K1-Y3)*10000
2250 LET K2=Y3
2260 GO TO 2390
2270 PRINT "DESIGNATE RANGE OF OLD BOND YIELDS AT TERMINAL DATE."
2280 PRINT "INPUT HIGH YIELD, LOW YIELD AND INCREMENT DESIRED,";
2285 PRINT " FOR EXAMPLE:"
2290 PRINT ".06,.05,.002."
2300 INPUT Z4,Z5,Z6
2310 LET Y6=Y4-Y3
2320REM
2330 FOR K2=Z5 TO Z4 STEP Z6
2340 LET K1=K2+Y6
2350 LET Z1=Z1+1
2360 LET A(Z1)=K2
2370 GO TO 2390
2380 GO TO 2400
2385REM      TIME TO SALE DATE
2390 LET S=S1*2+S2/6
2395REM      TIME TO MATURITY IN HALF YEARS
2400 LET M=M1*2+M3/6
2405REM  ACCRUED COUPON AT PRESENT
2410 LET A1=C1*(1-(M-INT(M)))
2420 LET N4=INT(M-S)
2425REM     TIME FROM PROPOSED SALE DATE TO THE NEXT COUPON
2430 LET N5=M-S-N4
2435REM    SEMI-ANNUAL COUPON
2440 LET C=C1
2445REM    SEMI-ANNUAL YIELD OLD BOND AT SALE DATE
2450 LET I=K2/2
2455REM     REDEMPTION PRICE OLD BOND
2460 LET A=R1
2470 GOSUB 4360
2480 LET A3=A9
2490 LET P3 = P
2495REM    NUMBER OF PERIODS THAT AMORTIZATION WAS TAKEN
2500 LET C7=INT(M)-INT(M-S)
2505REM   BOOK VALUE OLD BOND AT SALE DATE
2510 LET B3=B1-D1*C7
2520 GO TO 2570
2530 PRINT
```

```
2540 PRINT "PRICE OF OLD BOND AT SALE DATE";P3
2550 PRINT "ACCRUED INTEREST                              ",A3
2560 PRINT "BOOK VALUE AT SALE DATE                ",B3
2570 REM
2580 REM DATA ON REPURCHASED OLD BOND AT SALE DATE
2590 REM
2595REM    SEMI-ANNUAL PERIODS TO MATURITY  OLD BOND
2600 LET M=M1*2+M3/6
2605REM      ACCRUED COUPON AT PRESENT
2610 LET A1=C1*(1-(M-INT(M)))
2615REM   NUMBER OF COUPONS AFTER SALE
2620 LET N4=INT(M-S)
2625REM   TIME FROM PROPOSED SALE DATE TO NEXT COUPON
2630 LET N5=M-S-N4
2635REM       SEMI-ANNUAL COUPON
2640 LET C=C1
2645REM   ESTIMATED SEMI-ANNUAL YIELD OLD BOND AT SALE DATE
2650 LET I=K2/2
2655REM     REDEMPTION PRICE OLD BOND
2660 LET A=R1
2670 GOSUB 4360
2680 LET A3=A9
2690 LET P3=P
2695REM   NUMBER OF PERIODS THAT AMORTIZATION WAS TAKEN
2700 LET C7=INT(M)-INT(M-S)
2705REM      BOOK VALUE AT PROPOSED SALE DATE
2706REM
2710 LET B5=P1-D5*C7
2720 GO TO 2770
2730 PRINT
2740 PRINT"PRICE OF REPURCHASED OLD BOND AT SALE DATE";P3
2750 PRINT"ACCRUED INTEREST                              ",A3
2760 PRINT"BOOK VALUE AT SALE DATE                ",B5
2770 REM
2780 REM DATA ON NEW BOND AT SALE DATE
2790 REM
2795REM     SEMI-ANNUAL PERIODS TO MATURITY
2800 LET M=M2*2+M4/6
2805REM     ACCRUED COUPON AT PRESENT
2810 LET A2=C2*(1-(M-INT(M)))
2815REM    NUMBER OF COUPONS AFTER SALE TO MATURITY
2820 LET N4=INT(M-S)
2825REM   TIME FROM PROPOSED SALE DATE TO NEXT COUPON
2830 LET N5=M-S-N4
2835REM        SEMI-ANNUAL COUPON
2840 LET C=C2
2845REM   ESTIMATED SEMI-ANNUAL YIELD AT SALE DATE
2850 LET I=K1/2
2855REM  REDEMPTION PRICE NEW BOND
2860 LET A=R2
2870 GOSUB 4360
```

```
2880 LET A4 = A9
2890 LET P4 = P
2895REM   NUMBER OF PERIODS THAT AMORTIZATION WAS TAKEN
2900 LET C8=INT(M)-INT(M-S)
2905REM     BOOK VALUE OF NEW BOND AT SALE DATE
2910 LET B4=P2-D2*C8
2920 GO TO 2980
2930 PRINT
2940 PRINT "PRICE OF NEW BOND AT SALE DATE";P4
2950 PRINT "ACCRUED INTEREST             ";A4
2960 PRINT "BOOK VALUE AT SALE DATE              ";B4
2970 PRINT
2980 REm
2990 REm CAP GAIN TAX AT SALE OF OLD BOND NOW
3000 REm
3005REM     PRICE OLD BOND NOW
3010 LET P9=P1
3015REM     BOOK VALUE OLD BOND NOW
3020 LET B9=B1
3025REM    CAPITAL LOSS RATE NOW
3030 LET T8=T3
3035REM    CAPITAL GAIN RATE NOW
3040 LET T9=T4
3050 GOSUB 4450
3055REM     CAPITAL GAIN OR LOSS NOW A/T
3060 LET G1=G
3070 REm
3080 REm CAP GAIN SALE AT SALE DATE OF OLD BOND
3090 REm
3095REM    PRICE OLD BOND AT SALE DATE
3100 LET P9=P3
3105REM    BOOK VALUE OLD BOND AT SALE DATE
3110 LET B9=B3
3115REM    CAPITAL LOSS RATE AT SALE DATE
3120 LET T8=T5
3125REM    CAPITAL GAIN RATE AT SALE DATE
3130 LET T9=T6
3140 GO SUB 4450
3145REM    CAPITAL GAIN OR LOSS AT SALE DATE A/T
3150 LET G3=G
3160 REM
3170 REM CAPITAL GAIN AT SALE DATE OF REPURCHASED OLD BOND
3180 REM
3185REM     PRICE OLD BOND AT SALE DATE
3190 LET P9=P3
3195REM     BOOK VALUE OLD BOND AT SALE DATE
3200 LET B9=B5
3205REM    CAPITAL LOSS RATE AT SALE DATE
3210 LET T8=T5
3215REM    CAPITAL GAIN RATE AT SALE DATE
3220 LET T9=T6
```

```
3230 GOSUB 4450
3235REM   CAPITAL GAIN OR LOSS AT SALE DATE A/T
3240 LET G9=G
3250 REM
3260 REm CAP GAIN SALE AT SALE DATE OF NEW BOND
3270 REm
3275REM    PRICE OF NEW BOND AT SALE DATE
3280 LET P9=P4
3285REM   BOOK VALUE OF NEW BOND AT SALE DATE
3290 LET B9=B4
3295REM   CAPITAL LOSS TAX RATE AT SALE DATE
3300 LET T8=T5
3305REM    CAPITAL GAIN TAX RATE AT SALE DATE
3310 LET T9=T6
3320 GOSUB 4450
3325REM   CAPITAL GAIN OR LOSS AT SALE DATE--AFTER TAX
3330 LET G4=G
3340 REm
3350 REm SALE OF OLD BOND NOW AND PURCHASE OF NEW
3360 REm
3365REM   AFTER TAX PROCEEDS SALE OF OLD BOND NOW
3370 LET H1=P1-G1+A1*(1-T1)
3375REM   A/T PROCEEDS SALE OF OLD BOND AT SALE DATE
3380 LET H3=P3-G3+A3*(1-T1)
3390   LET H5=P3-G9+A3*(1-T1)
3395REM   A/T PROCEEDS SALE OF PURCHASED NEW BOND
3400 LET H4=P4-G4+A4*(1-T2)
3410 REM
3420 REM AFTER TAX YIELD FROM KEEPING OLD BOND
3430 REM
3435REM     TIME TO MATURITY OLD BOND
3440 LET M=M1*2+M3/6
3450 LET K=0
3455REM     FRACTION OF PERIOD FROM SALE DATE TO NEXT COUPON
3460 LET M7=M-S-INT(M-S)
3465REM    COUPONS RECEIVED FROM NOW TO SALE DATE OLD BOND
3470 LET M8=C7-1
3475REM     FRACTION OF PERIOD FROM NOW TO NEXT COUPON
3480 LET M9=M-INT(M)
3485REM      AFTER TAX COUPON OLD BOND
3490 LET C=C3
3495REM      MARKET VALUE OLD BOND  NOW
3500 LET U1=P1+A1*(1-T1)
3505REM A/T PROCEEDS FROM SALE OF OLD BOND AT SALE DATE
3510 LET U2=H3
3520 GOSUB 4530
3525REM   ANNUAL YIELD TO SALE DATE  OLD BOND
3530 LET Y5=2*(EXP(Y)-1)
3540 LET L(Z1,K)=((INT(Y5*10000+.5))/100)
3550 LET W3=0
3560 FOR I= 1 TO M8
```

```
3565REM   COUPONS COMPOUNDING AT A/T APPORTUNITY RATE
3570 LET W3=(W3+C)*(1+Y9/2)
3580 NEXT I
3585 IF M8=-1 THEN 3600
3587REM   INTEREST FROM SALE DATE TO NEXT COUPON
3590 LET W3=(W3+C)*(1+Y9/2)↑(1-M7)
3595REM     TERMINAL VALUE OLD BOND AT SALE DATE
3600 LET H9=H3+W3
3610 LET K=K+3
3620 LET L(Z1,K)=H9
3630 GO SUB 4300
3640 REM
3650 REM AFTER TAX YIELD FROM TAX SWITCH
3660 REm
3665REM   NUMBER OF OLD BONDS REPURCHASED
3670 LET N1=H1/(P1+A1*(1-T1))
3675REM   TERMINAL VALUE REPURCHASED OLD BONDS
3680 LET H5=H5*N1
3685REM   TIME TO MATURITY OLD BOND


3690 LET M=N1*2+M3/6
3695REM   FRACTION OF PERIOD FROM SALE TO NEXT COUPON
3700 LET M7=M-S-INT(M-S)
3705REM     NUMBER OF COUPONS RECEIVED
3710 LET M8=C7-1
3715REM   FRACTION OF PERIOD FROM NOW TO NEXT COUPON
3720 LET M9=M-INT(M)
3725REM     A/T AND N1 ADJUSTMENT COUPON
3730 LET C=N1*C5
3735REM     MARKET VALUE OF OLD BOND NOW
3740 LET U1=P1+A1*(1-T1)
3745REM     TERMINAL PROCEEDS FROM SALE OF N1 OLD BONDS
3750 LET U2=H5
3760 GOSUB 4530
3765REM     ANNUAL YIELD TO SALE DATE FROM T-S
3770 LET Y7=2*(EXP(Y)-1)
3780 LET K=K-2
3790 LET L(Z1,K)=((INT(Y7*10000+.5))/100)
3800 LET W5=0
3810 FOR I= 1 TO M8
3815REM     COUPONS COMPOUNDING AT  A/T OPPURTUNITY RATE
3820 LET W5=(W5+C)*(1+(Y9/2))
3830 NEXT I
3835 IF M8=-1 THEN 3850
3837REM     INTEREST FROM SALE DATE TO NEXT COUPON
3840 LET W5=(W5+C)*(1+(Y9/2))↑(1-M7)
3845REM   TERMINAL VALUE REPURCHASED OLD BONDS AT SALE DATE
3850 LET H7=H5+W5
3860 LET K=K+3
3870 LET L(Z1,K)=H7
```

```
3880 GO SUB 4300
3890 REM
3900 REm AFTER TAX YIELD FROM SWITCH
3910 REm
3915REM    NUMBER OF NEW BONDS PURCHASED
3920 LET N2=H1/(P2+A2*(1-T2))
3930 LET H4=H4*N2
3935REM   TIME TO MATURITY NEW BONDS
3940 LET M=N2*2+M4/6
3945REM    FRACTION OF PERIOD FROM SALE TO NEXT COUPON
3950 LET M7=M-S-INT(M-S)
3955REM    COUPONS RECEIVED FROM NOW TO SALE DATE   NEW BOND
3960 LET M8=C8-1
3965REM FRACTION OF PERIOD TO NEXT COUPON   NEW BOND
3970 LET M9=M-INT(M)
3975REM     AFTER TAX COUPON FOR N2   NEW BONDS
3980 LET C=N2*C4
3990 LET U1=P1+A1*(1-T1)
3995REM    TERMINAL PROCEEDS FROM SALE OF N2 NEW BONDS
4000 LET U2=H4
4010 GOSUB 4530
4015REM    YIELD FROM SWITCH INTO N2   NEW BONDS
4020 LET Y8=2*(EXP(Y)-1)
4030 LET K=K-2
4040 LET L(Z1,K)=((INT(Y8*10000+.5))/100)
4050 LET W4=0
4060 FOR I= 1 TO M8
4065REM    COMPOUNDING COUPONS AT A/T OPPORTUNITY RATE
4070 LET W4=(W4+C)*(1+(Y9/2))
4080 NEXT I
4085 IF M8=-1 THEN 4100
4087REM      INTEREST FROM SALE DATE TO NEXT COUPON
4090 LET W4=(W4+C)*(1+(Y9/2))↑(1-M7)
4095REM      TERMINAL VALUE PURCHASED NEW BONDS AT SALE DATE
4100 LET H6=W4+H4
4110 LET K=K+3
4120 LET L(Z1,K)=H6
4130 GOSUB 4300
4140 ON Z+1 GO TO 4720,4270,4230,4180,4720
4180 NEXT K2
4190 LET K2=Y3
4200 LET Z8=Z1
4210 GO TO 4290
4220 LET A(Z1)=A(Z1)*10000
4230 NEXT K1
4240 LET K1=Y4
4250 LET Z8=Z1
4260 GO TO 4290
4270 NEXT S
4280 LET Z8=Z1
4290 GO TO 4870
```

```
4300 REM ROUNDING ROUTINE
4310 LET L(Z1,K)=INT(L(Z1,K)*100+.5)/100
4320 RETURN
4330 REm
4340 REm BOND PRICE SUBROUTINE
4350 REm
4355REM      PRESENT VALUE OF COUPONS TO MATURITY
4360 LET U1=C*(1-(1+I)↑(-N4))/I
4365REM    PRESENT VALUE OF MARKET PRICE OF BONDS
4370 LET U2=A/(1+I)↑N4
4375REM    RATE OF FRACTION OF PERIOD TO NEXT COUPON
4380 LET Z9=(1+I)↑N5-1
4385REM
4386REM     PRICE OF BOND IN THE FUTURE
4390 LET P=((U1+U2+C)/(1+Z9))-C*(1-N5)
4395REM          ACCRUED INTEREST
4400 LET A9=C*(1-N5)
4410 RETURN
4420 REm
4430 REm CAP GAIN OR LOSS TAX CALCULATION
4440 REm
4445REM     IS MARKET PRICE GREATER THAN BOOK VALUE
4450 IF P9>B9 THEN 4480
4455REM   CAPITAL LOSS TAX SAVINGS
4460 LET G=T8*(P9-B9)
4470 GO TO 4490
4475REM         CAPITAL GAIN TAX
4480 LET G=T9*(P9-B9)
4490REM
4500 REm
4510REM   YIELD CALCULATION USING FISCHER'S ALGORITHM
4520 REm
4530 LET Y = .0001
4540 GOSUB 4610
4550 LET D=P-(U1+U2)
4560 IF ABS(D/U3)<.00000005 THEN 4590
4565REM        NEW TRIAL INTEREST RATE
4570 LET Y=Y-D/U3
4580 GO TO 4540
4590 RETURN
4600 REm
4610 REm YIELD SUBROUTINE
4620 REm
4625REM        PROCEEDS FROM SALE OF OLD BOND NOW
4630 LET P=U1
4640 LET U1=U3=0
4645REM    PRESENT VALUE OF PROCEEDS FROM FUTURE SALE
4650 LET U2=U2*EXP(Y)↑(-M9-M8+M7-1)
4660 FOR J=0 TO M8
4665REM   PRESENT VALUE OF INCOME STREAM
4670 LET U1=U1+C*(EXP(Y))↑(-J-M9)
```

```
4675REM    FISCHER'S DURATION
4680 LET U3=U3+C*(J+M9)*EXP(Y)↑(-J-M9)
4690 NEXT J
4700 LET U3=U3+U2*(M8+1+M9-M7)
4710 RETURN
4720 PRINT
4730 PRINT
4740 PRINT"YIELD TO SALE DATE A/T IF OLD BOND KEPT",
4750 PRINT ((INT(100000*Y5+.000005))/100000)*100;
4755 PRINT TAB(52);"PERCENT"
4760 PRINT"YIELD TO SALE DATE A/T ON TAX SWITCH",
4770 PRINT ((INT(100000*Y7+.000005))/100000)*100
4780 PRINT"YIELD TO SALE DATE A/T ON SWITCH TO NEW BD",
4790 PRINT ((INT(100000*Y8+.000005))/100000)*100
4800 PRINT
4810 PRINT
4820 PRINT "TERMINAL VALUE IF OLD BOND KEPT",((INT(H9*100+.5))/100)
4830 PRINT "TERMINAL VALUE OF TAX SWITCH     ",((INT(H7*100+.5))/100)
4840 PRINT "TERMINAL VALUE OF SWITCH TO NEW BOND",((INT(H6*100+.5))/100)
4850 PRINT
4860 GO TO 5180
4870REM
4880 PRINT
4890 ON Z+1 GO TO 4910,4960,5000,4910,4910
4910 LET D$="YIELD"
4920 LET D1$="(OLD BOND)"
4930 PRINT "SENSITIVITY OF YIELD AND TERM. VALUE TO ";D$;" AT TERM."
4940 PRINT "SPREAD HELD CONSTANT."
4950 GO TO 5040
4960 LET D$="TERM. DATE"
4970 LET D1$="(IN YEARS)"
4980 PRINT "SENSITIVITY OF YIELD AND TERM. VALUE TO ";D$;" AT TERM."
4990 GO TO 5040
5000 LET D$="SPREAD"
5010 LET D1$="(IN BASIS PTS.)"
5020 PRINT "SENSITIVITY OF YIELD AND TERMINAL VALUE TO ";D$;" AT TERM."
5030 PRINT "OLD BOND YIELD HELD CONSTANT."
5040 PRINT
5050 PRINT D$;TAB(13);"YIELD TO TERMINAL DATE";TAB(43);"TERMINAL VALUE"
5060 PRINT D1$
5070 PRINT
5080 PRINT TAB(10);"OLD BOND";TAB(20);"TAX SW.";TAB(30);"NEW BOND";
5090 PRINT TAB(40);"OLD BOND";TAB(50);"TAX SW.";TAB(60);"NEW BOND"
5100 FOR I=1 TO Z8
5110 PRINT A(I);
5120  FOR J=0 TO 5
5130 PRINT TAB(10*J+10);L(I,J);
5140 NEXT J
5150 PRINT
5160 NEXT I
5170 PRINT
```

```
5180 GO TO1970
5190 PRINT "DO YOU WANT A SUMMARY OF BOND DATA ";
5200 INPUT D$
5210 IF D$="NO" THEN 5620
5220 PRINT
5230 PRINT
5240 PRINT "SUMMARY OF DATA"
5250 PRINT
5260 PRINT TAB(26);"YIELDS TO MATURITY"
5270 PRINT
5280 PRINT "YIELD ON OLD BOND BEFORE TAX              ",
5290 PRINT ((INT(Q1*10000+.5))/100)
5300 PRINT "YIELD ON OLD BOND AFTER TAX                 ",
5310 PRINT ((INT(Q3*10000+.5))/100)
5320 PRINT "YIELD ON REPURCHASED OLD BOND AFTER TAX",
5330 PRINT ((INT(Q5*10000+.5))/100)
5340 PRINT "YIELD ON NEW BOND BEFORE TAX              ",
5350 PRINT ((INT(Q2*10000+.5))/100)
5360 PRINT "YIELD ON NEW BOND AFTER TAX               ",
5370 PRINT ((INT(Q4*10000+.5))/100)
5380 PRINT
5390 PRINT "OLD BOND"
5400 PRINT "---------"
5410 PRINT
5420 PRINT "PRESENT PRICE";P1,"COUPON";2*C1
5430 PRINT "REDEMPTION AT";R1,"BOOK VALUE";B1
5440 PRINT "MATURITY",M1;"YEARS",M3;"MONTHS"
5450 PRINT "TAX RATES",T1;"ON INC",T4;"ON CAP GNS"
5460 PRINT T3;"ON CAP LOSSES",D3;"ON CAP GNS AT MATURITY"
5470 PRINT "YIELD BEF TAX FORECAST FOR SALE DATE",Y3
5480 PRINT
5490 PRINT "NEW BOND"
5500 PRINT "---------"
5510 PRINT
5520 PRINT "PRESENT PRICE";P2,"COUPON";2*C2
5530 PRINT "REDEMPTION AT";R2
5540 PRINT "MATURITY",M2;"YEARS",M4;"MONTHS"
5550 PRINT "TAX RATES",T2;"ON INC",D4;"ON CAP GNS AT MATURITY"
5560 PRINT "YIELD BEF TAX FORECAST FOR SALE DATE",Y4
5570 PRINT
5580 PRINT
5590 PRINT "SALE DATE IS IN",S1;"YEARS",S2;"MONTHS"
5600 PRINT "TAX",T5;"ON CAP LOSS",T6;"ON CAP GNS"
5610 PRINT "OPPORTUNITY RATE A/T",Y9
5620 END
```

EXAMPLE

In this example problem the investor is looking ahead 2 years in endeavoring to choose between keeping a 2-3/4% debenture bond maturing in 1972 or switching to a 5-1/8% debenture bond maturing in 2001. The investor has entered the following data for the old and new bonds:

	OLD BOND	NEW BOND
Book Value (Old Bond)	$90-3/8	
Current Price	$91.00	$84.00
Annual Coupon	2-3/4%	5-1/8%
Maturity (Yrs./Mths.)	2-Yrs./5-Mths.	32-Yrs./9-Mths.
Redemption Price	$100.00	$100.00
Tax Rate on Coupon Int.	48%	48%
Terminal Date (Yrs./Mths.)	2-Yrs./0-Mths.	2-Yrs./0-Mths.
Yield at Terminal Date (Est.)	5%	4-1/2%
Capital Loss Rates at:		
Present Date	25%	25%
Terminal Date	25%	25%
Capital Gain Rates at:		
Present Date	25%	25%
Terminal Date	25%	25%
Maturity Date	25%	25%
After Tax Reinvestment Rate		
for Coupon Int. (Est.)	3%	3%

The redemption price shown includes any call premium and the reinvestment rate for coupon interest is per year up to the terminal date. The data may be input using DATA statements or entered in response to program prompting. In the following illustration data has been entered using DATA statements which contain all of the above information.

```
DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA ?NO
ENTER DATA ITEMS 1 THROUGH 13

?90.375,91,2.75,100,.48,84,5.125,32,9,100,.48

ENTER DATA ITEMS 14 THROUGH 24

?2,0,.05,.045,.25,.25,.25,.25,.25,.25,.03

TABLE OF YIELDS AND TERMINAL VALUES AT TERM. DATE TYPE:
    0-FOR NO TABLES
    1-FOR SENSITIVITY OF YIELD AND TERMINAL VALUES TO TERM.
    2-FOR SENSITIVITY TO YIELD SPREAD
    3-FOR SENSITIVITY TO YIELDS
    4-FOR YIELDS AND TERMINAL VALUES AT YOUR TERM. DATE

WHICH TABLE DO YOU WANT ?1
```

The example below illustrates how problem data is entered in response to program prompting.

```
*RUN

DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA ?YES

ENTER THE FOLLOWING ITEMS OF DATA

    ITEM    1       BOOK VALUE OF OLD BOND          ?90.375

    ITEM    2       PRICE OF OLD BOND               ?91

    ITEM    3       COUPON ON OLD BOND              ?2.75

    ITEMS   4 &  5  MATURITY OLD BOND IN YRS,MTHS.  ?2,5

    ITEM    6       PROCEEDS ON REDEMPTION OLD BOND ?100

    ITEM    7       TAX RATE ON OLD BOND INTEREST PMTS ?.40

    ITEM    8       PRICE OF NEW BOND               ?84

    ITEM    9       COUPON ON NEW BOND              ?5.125

    ITEMS  10 & 11  MATURITY NEW BOND IN YRS,MTHS.  ?32,9

    ITEM   12       PROCEEDS ON REDEMPTION NEW BOND ?100

    ITEM   13       TAX RATE ON NEW BOND INTEREST PMTS ?.40

    ITEMS  14 & 15  YRS,MTHS TO TERMINAL DATE       ?2,0

    ITEM   16       PREDICT YIELD OLD BOND TERM. DATE ?.05

    ITEM   17       PREDICT YIELD NEW BOND TERM. DATE ?.045

    ITEMS  18 & 19  CAP LOSS,GAIN RATES NOW         ?.25,.25

    ITEMS  20 & 21  CAP LOSS,GAIN RATES TERM. DATE  ?.25,.25

    ITEM   22       CAP GAIN RATE AT MATURITY OLD BOND ?.25

    ITEM   23       CAP GAIN RATE AT MATURITY NEW BOND ?.25

    ITEM   24       AFTER TAX REINVEST RATE COUPONS   ?.03
```

After the initial problem data has been entered the program will ask the user
which of 5 reports is wanted. By typing a Ø to the question "Which Table Do
You Want?" and a "YES" to the next question the user will be able to verify
his data input.

```
DO YOU WANT INSTRUCTIONS FOR SUPPLYING DATA ?DATA
TABLE OF YIELDS AND TERMINAL VALUES AT TERM. DATE. TYPE:
    0-FOR NO TABLES
    1-FOR SENSITIVITY OF YIELDS AND TERMINAL VALUES TO TERM.
    2-FOR SENSITIVITY TO YIELD SPREAD
    3-FOR SENSITIVITY TO YIELDS
    4-FOR YIELDS AND TERMINAL VALUES AT YOUR TERM. DATE

WHICH TABLE DO YOU WANT ?0

DO YOU WANT A SUMMARY OF BOND DATA ?YES


SUMMARY OF DATA

                              YIELDS TO MATURITY

YIELD ON OLD BOND BEFORE TAX                          6.85
YIELD ON OLD BOND AFTER TAX                           4.44
YIELD ON REPURCHASED OLD BOND AFTER TAX               4.51
YIELD ON NEW BOND BEFORE TAX                          6.28
YIELD ON NEW BOND AFTER TAX                           3.41

OLD BOND

PRESENT PRICE            91              COUPON        2.75
REDEMPTION AT            100             BOOK VALUE  90.375
MATURITY          2 YEARS      5 MONTHS
TAX RATES         .48 ON INC       .25 ON CAP GNS
          .25 ON CAP LOSSES       .25 ON CAP GNS AT MATURITY
YIELD BEFORE TAX FORECAST FOR SALE DATE               .05

NEW BOND

PRESENT PRICE            84              COUPON       5.125
REDEMPTION AT            100
MATURITY          32 YEARS      9 MONTHS
TAX RATES         .48 ON INC       .25 ON CAP GNS AT MATURITY
YIELD BEFORE TAX FORECAST FOR SALE DATE               .045

SALE DATE IS IN             2 YEARS        0 MONTHS
TAX         .25 ON CAP LOSS            .25 ON CAP GNS
OPPORTUNITY RATE A/T              .03
```

677

The first table "Sensitivity of Yield & Terminal Values to Term." illustrates that for the 2 year future time period a switch for tax purposes would not be to the investor's advantage, however there would be a substantial advantage to the investor in switching in the new bond's rising terminal value. As the dollar advantage in terminal value is roughly the same whether the yield forecasts are achieved in 1/2 year or in 3 years, the terminal advantage of the new bond, if earned quickly (yield comparisons show early achievement) would prove to make the switch even more profitable.

SENSITIVITY OF YIELD AND TERMINAL VALUES TO TERM. DATE AT TERM.

TERM. DATE (IN YEARS)	YIELD TO TERMINAL DATE			TERMINAL VALUE		
	OLD BOND	TAX SW.	NEW BOND	OLD BOND	TAX SW.	NEW BOND
.5	9.36	9.34	50.17	96.11	96.1	115.29
1.0	6.34	6.23	25.22	96.87	96.86	115.22
1.5	5.21	5.2	17.47	98.4	98.39	116.65
2.0	4.7	4.7	13.69	99.96	99.94	118.09
2.5	4.4	4.4	11.46	101.54	101.53	119.56
3.0	4.2	4.2	9.99	103.16	103.14	121.05

After the print out of the first table the user is once again queried as to which table he wants. In this second illustration the user has responded by typing a 2, calling for the "Sensitivity of Yield & Terminal Value to Spread at Term." table. Additional information is required for this table and the user in response to program prompting has supplied the range of spreads over which the analysis is to be made. In the example the spread ranges from -100 basis points to +100 basis points with the results to be shown in increments of 30 basis points. The table illustrates that the more negative the spread at the terminal date, the more profitable the switch will be. To determine at exactly what point a switch would cease to be profitable, rerun the program using different spreads. Note: The table uses the old bond's original yield forecast while the yield forecast for the new bond is varied in increments of 30 basis points.

WHICH TABLE DO YOU WANT ?2

DESIGNATE RANGE OF SPREAD (NEW BOND YIELD-OLD BOND YIELD) AT
TERMINAL DATE; INPUT HIGH SPREAD, LOW SPREAD AND INCREMENT
IN BASIS POINTS. FOR EXAMPLE, -100,104,25.
?-100,100,30

SENSITIVITY OF YIELD AND TERMINAL VALUE TO SPREAD AT TERM.
OLD BOND YIELD HELD CONSTANT

SPREAD (IN BASIS PTS.)	YIELD TO TERMINAL DATE			TERMINAL VALUE		
	OLD BOND	TAX SW.	NEW BOND	OLD BOND	TAX SW.	NEW BOND
-100.00000	4.7	4.7	17.13	99.96	99.94	125.71
-70.00001	4.7	4.7	15.04	99.96	99.94	121.03
-40.00001	4.7	4.7	13.03	99.96	99.94	116.67
-10.00002	4.7	4.7	11.11	99.96	99.94	112.61
19.99998	4.7	4.7	9.26	99.96	99.94	108.83
49.99998	4.7	4.7	7.49	99.96	99.94	105.29
79.99997	4.7	4.7	5.8	99.96	99.94	101.99

In this illustration the investor has requested a print out of table #3, the "Sensitivity of Yield & Terminal Value to Yield at Term." In response to program prompting the investor has specified a high yield of 6% and a low yield of 5% (original estimate) for the old bond at the terminal date and has indicated he would like to see the comparisons made in increments of 20 basis points. As the table shows, if the old bond is yielding 6% at the terminal date rather then the 5% originally estimated by the investor a switch would still prove to be profitable, but it would be considerably less advantageous. Note: For this illustration the yield spread has been held constant at the original -50 basis points already entered in the program data.

WHICH TABLE DO YOU WANT ?3

DESIGNATE RANGE OF OLD BOND YIELDS AT TERMINAL DATE.
INPUT HIGH YIELD, LOW YIELD AND INCREMENT DESIRED,
FOR EXAMPLE: .06,.05,.002.
?.06,.05,.002

SENSITIVITY OF YIELD AND TERM. VALUE TO YIELD AT TERM.
SPREAD HELD CONSTANT.

YIELD (OLD BOND)	YIELD TO TERMINAL DATE			TERMINAL VALUE		
	OLD BOND	TAX SW.	NEW BOND	OLD BOND	TAX SW.	NEW BOND
.05	4.7	4.7	13.69	99.96	99.94	118.09
.052	4.67	4.66	12.38	99.9	99.88	115.29
.054	4.64	4.63	11.11	99.84	99.82	112.61
.056	4.61	4.6	9.87	99.77	99.76	110.06
.058	4.58	4.57	8.66	99.71	99.7	107.62
.06	4.55	4.54	7.49	99.65	99.64	105.29

Finally, the investor responds by typing a 4 to the question "Which Table Do You Want?". This table of "Yields & Terminal Values at Terminal Date" uses the data originally supplied to print out a table of yields and terminal values as of the terminal date.

```
WHICH TABLE DO YOU WANT ?4

YIELD TO SALE DATE A/T IF OLD BOND KEPT              4.783  PERCENT
YIELD TO SALE DATE A/T ON TAX SWITCH                 4.695
YIELD TO SALE DATE A/T ON SWITCH TO NEW BOND        12.689

TERMINAL VALUE IF OLD BOND KEPT                      99.96  DOLLARS
TERMINAL VALUE OF TAX SWITCH                         99.94
TERMINAL VALUE OF SWITCH TO NEW BOND                118.99
```

RELIABLE COMPUTER SOFTWARE

FOR YOUR DOWN TO EARTH TASKS